

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a
komunikačních technologií

BAKALÁŘSKÁ PRÁCE



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

3D MĚŘENÍ POMOCÍ ŘÁDKOVÝCH KAMER

3D RECONSTRUCTION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Daniel Kluka

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Miloslav Richter, Ph.D.

BRNO 2020

Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Daniel Kluka **ID:** 203251 **Ročník:** 3 **Akademický rok:** 2019/20

NÁZEV TÉMATU:

3D měření pomocí řádkových kamer

POKYNY PRO VYPRACOVÁNÍ:

Úkolem je vypracovat algoritmy pro měření 3D tvaru pohybujících se objektů pomocí řádkových kamer.

- 1) Nastudujte možnosti snímání obrazu pomocí řádkových kamer. Pro měření zvolte dvě sestavy a jejich výsledky porovnejte. V obou případech se kamery dívají ve scéně na společnou přímku. V prvním případě tvoří zobrazovací roviny písmeno V, ve druhém případě jsou v jedné rovině.
- 2) Navrhněte jednoduchý matematický aparát pro vyhodnocení 3D tvaru.
- 3) Realizujte testovací pracoviště a nasnímejte scény pro ověřování algoritmů.
- 4) Realizujte základní algoritmy měření a zhodnoťte dosažené výsledky. Srovnajte jednotlivé metody vůči sobě a popište jejich výhody a nevýhody.

DOPORUČENÁ LITERATURA:

Gonzalez R.C., Woods R.E.: Digital Image Processing, 4th edition, Pearson, 2017, ISBN 978-0133356724

Termín zadání: 3.2.2020

Termín odevzdání: 8.6.2020

Vedoucí práce: Ing. Miloslav Richter, Ph.D.

doc. Ing. Václav Jirsík, CSc.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Abstrakt

Bakalárska práca sa zaoberá tvorbou algoritmu na spracovanie obrazu pomocou riadkovej kamery. K úspešnému snímaniu predchádza tvorba algoritmu na korekciu sústavy, ktorý je potrebný pri spracovaní obrazu. Vývoj algoritmov prebieha vo vývojovom programovacom prostredí PyCharm spolu s programovacím jazykom Python a knižnicou OpenCV. Práca sa sčasti venuje spracovaniu obrazu aj vo vývojovom prostredí Matlab. Hlavným cieľom je naštudovať možnosti spracovania obrazu pomocou riadkových kamier a navrhnúť jednoduchý matematický aparát pre vyhodnotenie 3D tvaru.

Summary

The bachelor's thesis deals with the creation of algorithms for image processing of control machines. For successful scanning before creating an algorithm to correct the system that is during image processing. Development of algorithms running in the PyCharm development programming environment together with the Python programming language and the OpenCV library. The work is partly devoted to image processing in the Matlab development environment. The main goal is to study the possibilities of scanning control cameras and designing suitable mathematical devices before using a 3D shape.

Klíčová slova

3D meranie, riadková kamera, kalibrácia kamery, OpenCV, Python, epipolárna geometria

Keywords

3D measurement, line camera, camera calibration, OpenCV, Python, epipolar geometry

KLUKA, Daniel. 3D měření pomocí řádkových kamer. Brno, 2020. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/126971>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Miloslav Richter.

Čestne prehlasujem, že som bakalársku prácu „3D meranie pomocou riadkových kamier“ vypracoval samostatne s použitím literatúry uvedenej v zozname a na základe svojich vedomostí.

.....

podpis

Ďakujem pánovi profesorovi Ing. Miloslavovi Richterovi za jeho pomoc a čas, ktorý mi venoval formou konzultácií. Jeho podpora a odborné znalosti mi veľmi pomáhali pri pracovaní na tejto téme a vďaka nim som nestratil motiváciu skúmať nové veci a naučiť sa niečo nové. Ďakujem mojej rodine, môjmu bratovi a kamarátom, ktorí ma podporovali v štúdiu na tejto škole a stáli pri mne v tých najťažších chvíľach.

.....

podpis

Obsah

Úvod.....	9
2 Rešeršná štúdia	10
2.1 Spracovanie a vlastnosti obrazu	10
2.1.1 Jas a hĺbka obrazu.....	10
2.1.2 Kontrast a dynamický rozsah.....	10
2.1.3 Farebný model	10
2.2 Digitalizácia obrazu	12
2.2.1 Vzorkovanie.....	12
2.2.2 Kvantovanie.....	12
2.3 Videokamera	12
2.3.1 Obrazový snímač	13
2.3.2 Riadková kamera	13
2.4 Dierkový model kamery.....	14
2.5 Kalibrácia videokamery	16
2.5.1 Radiálne skreslenie	16
2.5.2 Tangenciálne skreslenie.....	17
2.5 Epipolárna geometria	17
2.5.1 Parametre kamery	18
2.5.2 Fundamentálna matica	18
3 Realizácia.....	19
3.1 Knižnica OpenCV	19
3.2 Programovací jazyk Python	19
3.3 Kalibrácia kamery	20
3.4 Simulácia riadkovej kamery.....	22
3.4.1 Zhodnotenie merania pomocou simulácie riadkovej kamery	22
3.5 Meranie pomocou reálnej riadkovej kamery.....	23
3.5.1 Porovnanie kamery so simuláciou	23
3.5.2 Výpočet správneho pomeru strán snímok (aspect ratio)	24
3.5.3 Detekcia hrán meraného objektu	26
3.5.4 Detekcia objektu pomocou plošnej kamery.....	27
3.6 Overovanie algoritmov na tvorbu 3D modelu	28
3.6.1 Porovnávacie algoritmy	28
3.6.2 Rektifikácia obrazu	29
3.6.3 3D model vytvorený pomocou hĺbkovej mapy.....	30
3.7 Návrh matematického modelu počítania 3D modelu.....	30
3.7.1 Určenie náklonu objektu.....	30
3.7.1 Korekcia skreslenia objektu v snímke z jednej kamery	32
3.7.2 Počítanie výšky objektu	34

3.8	Tvorba 3D modelu	37
3.8.1	Postup pri tvorbe algoritmu vykresľovania	37
3.8.2	Výsledné snímky 3D modelu.....	38
3.8.3	Zhodnotenie 3D modelu scény	39
3.9	Posuny dvoch snímok	41
3.9.1	Odstránenie výchylky rovín prvej snímky.....	41
3.9.2	Určenie rotácie a translácie prvej snímky	44
3.9.3	Odstránenie výchylky rovín druhej snímky.....	45
3.9.4	Určenie rotácie a translácie prvej snímky	45
3.9.5	Výsledné korekcie rovín dvoch snímok.....	46
4	Výsledky a záver	47
5	Zoznam použitej literatúry.....	48
	Prílohy.....	49

Úvod

Zrak je jeden zo zmyslov človeka, pomocou ktorého dokážeme identifikovať farby, tvary a rôzne objekty. Aj napriek tomu, že u každej osoby sa zrak líši, človek je schopný rozoznať 3D tvar objektu. V dnešnej dobe dokážu takto spracovať obraz rôzne prístroje pomocou kamier a fotoaparátov. Najčastejšie sa stretáme s 3D vo svete filmu, kedy sme pomocou okuliarov schopní vidieť obraz viacrozmerne. V oblasti priemyslu sa pomocou kamier a počítačového videnia analyzuje kvalita produktov, ktorú by človek nedokázal analyzovať tak, ako stroj. S odstupom času sa spracovanie obrazu posunulo na inú úroveň – na tvorbu 3D objektov v reálnom svete vo fyzickej podobe. Toto je možné pomocou tzv. 3D tlačiarň a využíva sa na tvorbu rôznych pomôcok, častí do rôznych zariadení alebo v medicíne.

3D meranie obrazu pomocou riadkových kamier sa využíva v priemysle, kedy je potrebné zistiť kvalitu produktu a určiť chybný výrobok, poprípade doladiť stroj, ktorý daný produkt vyrába na základe 3D meraní. Dostupnosť týchto kamier je však obmedzená. Kvôli ich vysokým cenám sa tvoria simulácie s plošnými kamerami a pomocou algoritmov sa vlastnosťami stávajú plošné kamery riadkovými. Plošné kamery sa oproti riadkovým nastavujú lepšie, lepšie sa s nimi pracuje a výsledný obraz z plošnej kamery vieme upraviť tak, aby nám dal iba jeden riadok, rovnako ako na riadkových kamerách. Základom tohto merania je algoritmus, ktorý dokáže spracovať obraz a vytvoriť viacrozmerný model, vďaka ktorému by sme mohli analyzovať daný objekt. S týmto meraním sa spája aj Raspberry Pi alebo Arduino, ktoré je schopné daný algoritmus spracovávať a knižnica OpenCV pre rôzne programovacie jazyky, napríklad Python, vďaka ktorej sme schopní tento algoritmus vytvoriť. Ako riadková kamera nám poslúži video, resp. webkamera, na ktorých budeme meranie vykonávať. V spojení s touto témou je dôležitá aj kalibrácia kamery, aby sme dosiahli čo najpresnejšie spracovanie objektu a odstránili zakrivenie šošovky kamery.

3D je v dnešnej dobe výraz, s ktorým sa stretávame na každom rohu. Či už je to tlač, film, priemysel alebo náš zrak, sme schopní vytvárať pozoruhodné veci. Nakoľko mám rád filmy a ich neodmysliteľnou súčasťou sú kamery, tak ma táto téma nadchla a zaujala. Nielen z programátorského hľadiska, ale aj z hľadiska spracovania obrazu a tvorby objektov. Nahliadnutím do oblasti priemyslu a výroby prostredníctvom počítačového videnia mi táto téma môže poskytnúť nové skúsenosti v programovaní a priniesť mi nové zaujímavosti, ktoré ma vo svete filmu dokážu nadchnúť.

2 Rešeršná štúdia

2.1 Spracovanie a vlastnosti obrazu

Zrak je zmysel človeka, pomocou ktorého vieme rozoznávať farby a tvary rôznych objektov pomocou oka. Oko je optická sústava tvoriaca obraz, ktorý je priamy, prevrátený a zmenšený. Tvorí sa v ohniskovej rovine na sietnici pomocou perspektívnej projekcie. V matematike definujeme obrazovú funkciu $f(x, y)$, ktorú máme zloženú z priestorových súradníc x a y tvoriacich pixel. Hodnota tejto obrazovej funkcie je úmerná napríklad jas, optickej hustote pri priehľadných predlohách, vzdialenostiach od pozorovateľa, atď. Táto hodnota je veličina snímaná obrazovým snímačom – v počítačovom videní je to kamera, ktorá sníma obraz a prenáša ho do digitálnej podoby pomocou procesu digitalizácie. Výsledkom digitalizácie je digitálny obraz, popísaný digitálnou obrazovou funkciou odvodenou z obrazovej funkcie. Tento obraz je navzorkovaný v matici o rozmeroch $(M \times N)$, kde M je šírka a N je výška obrazu v pixeloch. Kvantovaním obrazu do K intervalov môžeme dosiahnuť lepšiu aproximáciu obrazovej funkcie, ak je rozlíšenie a kvantovanie jemnejšie. Informácie v tejto kapitole získané z [8], [4], [12], [15].

2.1.1 Jas a hĺbka obrazu

Jas je svietivosť farebného pixelu, charakterizuje celkovú jasnosť alebo tmavosť obrazu. Často sa spája so sýtosťou a miešaním rôznych farieb, napríklad miešaním čiernej a bielej farby môžeme sýtosť a jas meniť. Ľudské oko môže rozlíšiť až 1,5 milióna rôznych farieb, ktoré vníma ako svetlo, teda elektromagnetické vlnenie s rôznou vlnovou dĺžkou. Nie však každá farba má inú vlnovú dĺžku. Podľa intenzity svetla oko rozlišuje sýtosť a jas daného odtieňa farby. Čierna farba má hodnotu jasu 0, biela, naopak, najväčšiu použiteľnú hodnotu. Jas je úzko spätý s bitovou hĺbkou obrazu, ktorá udáva počet bitov, ktoré zaberú v pamäti číslo charakterizujúce jas jedného pixelu. Pri pohľade na obraz ako celok viacerých pixelov, jas určuje jasnosť alebo tmavosť obrazu.

2.1.2 Kontrast a dynamický rozsah

Kontrast je rozdiel medzi najsvetlejšími a najtmavšími oblasťami obrazu. Spomedzi rôznych metód zmien kontrastu sú najznámejšie *Low Key* a *High Key*. *Low Key* osvetlenie je štýl osvetlenia fotografickej scény, v ktorom prevažujú tmavé tóny nad svetlými. *High Key* je opak *Low Key*, kedy svetlé tóny prevažujú nad tmavými. Podiel najvyššieho a najnižšieho jasu daného formátu sa nazýva dynamický rozsah. Vlastnosť podobná kontrastu je farebná sýtosť, ktorá sa zameriava na farebné rozlíšenie oblastí. Kontrast sa sčasti využíva aj pri ostrosti obrazu – pri zvýraznení hrán objektu, resp. v ich blízkosti.

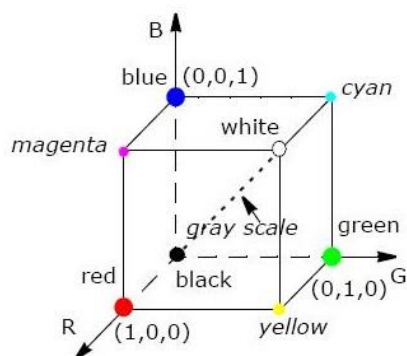
2.1.3 Farebný model

Farba pixelu sa reprezentuje pomocou čísel, ktoré reprezentujú jas farieb, z ktorých sa pixel skladá. Farba pixelu je reprezentovaná ako bod farebného priestoru, ktorý je umiestnený v počiatku trojrozmernej súradnicovej sústavy, v niektorých prípadoch až štvorrozmernej.

RGB model (Red - červená, Green - zelená, Blue - modrá)

Je najčastejšie používaný model, skladajúci sa z troch základných farieb – červenej (R), zelenej (G) a modrej (B). Pomocou farebného priestoru a súradnicovej sústavy vieme vytvoriť výslednú aditívnu farbu, ktorá je zložená z týchto troch farieb. Pomocou jednotkovej kocky vieme určiť súradnice jednotlivých farieb. Červená je na súradniciach (1,0,0), zelená na súradniciach (0,1,0) a súradnice modrej farby sú (0,0,1). RGB model, so súradnicami červenej, zelenej a modrej farby, je znázornený na obrázku č. 1. Tento model je využívaný napríklad v monitoroch a čím väčšia je hodnota farieb,

tým je výsledná farba svetlejšia. Miešaním farieb dokáže vytvoriť až 16 777 216 rôznych farieb, konkrétne až 256 odtieňov z každej farby (8 bitov).



Obr. č. 1: RGB model jednotkovej kocky (získaný z [2])

CMY(K) (Cyan, Magenta, Yellow, Key – black)

Tento model sa skladá z troch farieb – azúrová (C), fialová (M) a žltá (Y). Zatiaľ, čo RGB model využíva aditívne miešanie farieb, model CMY(K) využíva substraktívne miešanie. Najčastejšie je využívaný pri tlači a kvôli úspore troch hlavných farieb, sa používa aj informácia o čiernej zložke (K). Vrcholom jednotkovej kocky modelu CMY(K) je biela farba na súradniciach (1,1,1). V iných častiach kocky môžeme vidieť azúrovú farbu na súradniciach (1,0,0), fialovú farbu na súradniciach (0,1,0) a poslednú – žltú farbu na súradniciach (0,0,1).

HSB/HSV a HSL (Hue – odtieň, Saturation – sýtosť, Brightness/Value – jas/hodnota, Lightness – svetlosť)

Jedná sa o kódovanie farieb, ktoré berie celý rozsah vlnových dĺžok a rozdeľuje ho na niekoľko častí. Do pamäte počítača ukladá poradové číslo farby, ktorej prislúcha určitá vlnová dĺžka, jej sýtosť a jas. Farebné modely HSB, HSV a HSL uchovávajú informáciu o odtieni a sýtosti. Je medzi nimi však rozdiel – model HSB a HSV uchovávajú jas, ktorý je označovaný ako Brightness alebo Value. Čiernu farbu v tomto modeli získame tak, že jas nastavíme na nulu a bielu tak, že ho nastavíme na maximum. Nastavovanie týchto farieb je nezávislé od sýtosti. Model HSL uchováva svetlosť, ktorá je označovaná ako Lightness. V tomto modeli naopak pracujeme aj so sýtosťou. Na to, aby sme v ňom nastavili čiernu farbu, musí byť svetlosť aj sýtosť nastavená na 0. Bielu farbu nastavíme nastavením svetlosti aj sýtosti na maximálnu hodnotu. Hlavným nedostatkom tohto kódovania je náročnosť výroby svetla so zadanou vlnovou dĺžkou.

YUV (Y (luminance) – čiernobiela zložka, UV (chrominance) – farebnosť)

Tento farebný model zachováva čiernobiely informáciu pri televíznom vysielaní. Vznikol ako riešenie pri prechode z čiernobielych televízorov na farebné, kedy bol problém kódovať farebný obraz ako čiernobiely. Kódovanie spočíva v pridaní dvoch farebných zložiek farebnosti UV k čiernobielej zložke Y. Zložka U udáva odtieň medzi modrou a žltou, V udáva odtieň medzi červenou a žltou farbou. Čiernobiely prijímač berie do úvahy iba zložku Y, farebné prijímače dokážu získať RGB kód pomocou transformácie:

$$R = Y + 1,403 \cdot V \quad (2.1)$$

$$G = Y - 0,344 \cdot U - 0,714 \cdot V \quad (2.2)$$

$$B = Y + 1,770 \cdot U \quad (2.3)$$

2.2 Digitalizácia obrazu

Digitalizácia, je spracovanie analógového obrazu (analógovej informácie) do digitálnej podoby (číselný zápis) pomocou vzorkovania. Obraz definujeme pomocou súradníc x, y vo funkcii:

$$f(x, y) = H \quad (2.4)$$

Kde H je hodnota jas. Na digitalizáciu súradníc x, y a jas H je potrebná diskretizácia pomocou vzorkovania a kvantovania signálu. Informácie získané z [8], [5].

2.2.1 Vzorkovanie

Vzorkovanie, je proces, ktorý zo spojitkej funkcie $f(x, y)$ odoberá hodnoty (vzorky) v rovnakých intervaloch. Tento proces vzniká rozdelením vodorovnej osi na rovnaké časti. Z týchto častí sa potom vyberie pre každú časť jedna vzorka. Takto vznikne zo spojitkej čiary množina diskretných bodov. Body, ktoré sú v intervale, ktoré definuje vzorkovacia frekvencia f_{vz} :

$$f_{vz} = \frac{1}{\Delta x} \quad (2.5)$$

Kde Δx je rozmedzie dvoch vzoriek. V spojitosti so vzorkovacou frekvenciou je dôležité, aby vzorkovacia frekvencia bola dvojnásobne väčšia ako maximálna frekvencia, ktorá je použitá v obraze. Toto tvrdenie sa nazýva Shannonova veta a znamená, že interval vzorkovania musí mať takú veľkosť, aby bol menší alebo rovný polovici rozmeru najmenších detailov v obraze.

$$f_{vz} > 2f_{max} \quad (2.6)$$

2.2.2 Kvantovanie

Pomocou diskretizácie obrazovej funkcie vzniká aj proces zvaný kvantovanie. Každému intervalu z oboru hodnôt obrazovej funkcie sa pridružuje zástupná hodnota, ktorá sa získa pomocou priemeru alebo mediánu z hodnôt daného intervalu. Kvantovanie rozdeľujeme na dva typy – uniformné a neuniformné. Je medzi nimi iba jeden rozdiel, a to v dĺžke intervalu – uniformné používa konštantnú dĺžku intervalu a neuniformné premenlivú.

2.3 Videokamera

Kamera, je kamerové zariadenie na záznam pohyblivého obrazu so zvukom a jeho okamžité uloženie na záznamové médium vo vnútri kamery. Kamery rozdeľujeme podľa ich určenia na viacero druhov, napríklad televízna kamera, webová kamera, kamera vo smartfónoch, bezpečnostná kamera a pod. Snímanie obrazu je možné vďaka obrazovým snímačom. Dnešné moderné kamery zvládajú záznam v ultravysokom rozlíšení – 4096x2160 pixelov (poprípade viac) pri 24 až 120 snímkach za sekundu (fps – frames per second). Takýto záznam sa zväčša neskôr spracúva v štúdiu, napríklad aplikovaním špeciálnych efektov, strihom a pod. Výsledné dielo je prepracované na rozlíšenie a formát, ktoré sú domáce zariadenia, napríklad televízie, DVD alebo Blu-Ray prehrávače, satelitné prijímače schopné prehrať a zobraziť. Najrozšírenejším je rozlíšenie FullHD, čiže rozlíšenie 1920x1080 pixelov. S postupom času ale viaceré televízne stanice a satelitní operátori experimentujú s nasadením vyššieho rozlíšenia do televízie – je to rozlíšenie 4K, čiže 4096x2160 pixelov. Avšak takáto technika je veľmi pokročilá a drahá. Väčšina domácností má doma satelitné prijímače, ktoré vyššie rozlíšenie spracovať

nedokázu. Potrvá teda ešte nejaký čas, pokým 4K rozlíšenie nahradí už teraz bežné a overené rozlíšenie FullHD. Informácie v tejto kapitole získané z [14], [3], [7].

2.3.1 Obrazový snímač

Toto zariadenie sa skladá zo sústavy jednoduchých fotosenzorov, ako napríklad fotodióda, fotorezistor či fototranzistor a prevádza obrazovú informáciu na elektrický signál. Spočiatku sa na tento zámer používala elektrónka, ktorá zachytávala obraz a prevádzala ho na elektrický signál. Neskôr bola nahradená mikročipom na báze CCD, CMOS, NMOS alebo Live MOS.

Senzor CCD

Každá fotobunka CCD senzora je analógová – pasívna. Elektrický náboj sa vygeneruje až po dopade svetla na ňu a je reprezentovaný zmenou napätia na kontaktoch fotobunky. Toto napätie je merané ovládacím obvodom a je ďalej spracované. Výsledkom je digitálna informácia a elektronická podoba obrazu, ktorý dopadol na plochu celej sústavy fotobuniek senzora. CCD senzor je jednoduchý a lacný na výrobu. Jeho obmedzenie je najmä z hľadiska rýchlosti.

Senzor CMOS

Senzor CMOS má jednotlivé fotobunky aktívne. Jedná sa o polovodičovú technológiu. Každá fotobunka je fotodióda, cez ktorú prechádza elektrický impulz. Zmena v jeho veľkosti či v inom parametre nastáva v momente, keď na fotodiódu dopadne svetlo. Táto zmena je, podobne ako v senzore CCD, ďalej vyhodnocovaná a spracovávaná. Zložitejšia štruktúra samotného obrazového senzora znamená, že jeho výroba je náročnejšia a teda aj drahšia. Oproti senzoru CCD je v snímaní obrazu rýchlejší, čo býva kľúčový faktor pri zázname pohyblivého obrazu.

2.3.2 Riadková kamera

Riadková kamera obsahuje snímač, ktorý má v princípe iba jeden riadok pixelov a princípom funguje rovnako ako kancelársky skener. Pod kamerou sa lineárne pohybuje pás, ktorý kamera sníma riadok po riadku a z nich sa potom skladá dohromady obraz. Obraz je potom spracovaný softwarom. Takýto princíp kamery umožňuje snímanie pohyblivých objektov vysokou rýchlosťou s pomerne veľkými detailmi.

Výhody riadkovej kamery oproti plošnej/maticovej CCD kamere

Pri snímaní obrazu plošnou kamerou sa volí veľmi krátky expozičný čas a veľmi intenzívne zábleskové svetlo, aby sa vyhlo rozmazaniu obrazu vplyvom pohybu snímaného predmetu. Lineárny snímač je niekoľkonásobne citlivejší v porovnaní s plošným senzorom a preto môže snímať s naopak veľmi krátkym expozičným časom vysokou rýchlosťou v rádoch desiatok tisíc riadkov za sekundu. Pre nasvetlenie snímaného miesta postačí intenzívne líniové svetlo, ktoré sústreďuje všetok svetelný tok do úzkej linky. Z pohľadu optiky je riadková kamera lepším riešením, pretože vady a nepresnosti na objektíve sa prejavujú iba v jednom smere. Ďalšou výhodou je rozlíšenie, ktoré v súčasných riadkových kamerách dosahuje až 16 tisíc bodov. Pozdĺžne rozlíšenie závisí iba na frekvencii a dobe snímania kamery. Takéto parametre neponúka žiadna komerčne dostupná plošná kamera.

Parametre riadkovej kamery

Najjednoduchším modelom riadkovej je monochromatické riadková kamera poskytujúca jeden riadok pixelov. Jej senzor je vyrobený pokročilou technológiou CMOS s vysokou kvantovou účinnosťou pre dosiahnutie čo najlepšej citlivosti. Pixely senzora sú usporiadané tak, aby efektívne využívali celú plochu pixela. Takáto kamera ponúka rozlíšenie 2 až 12 tisíc obrazových bodov a vďaka použitej technológii snímača je možné

dosiahnuť maximálnu snímáciu frekvenciu až 80 kHz. Existujú aj ďalšie varianty riadkových kamier, ktoré obsahujú napríklad iné senzory, iný počet riadkov (napríklad dva alebo tri) alebo fungujú na iných princípoch spracovania obrazu.

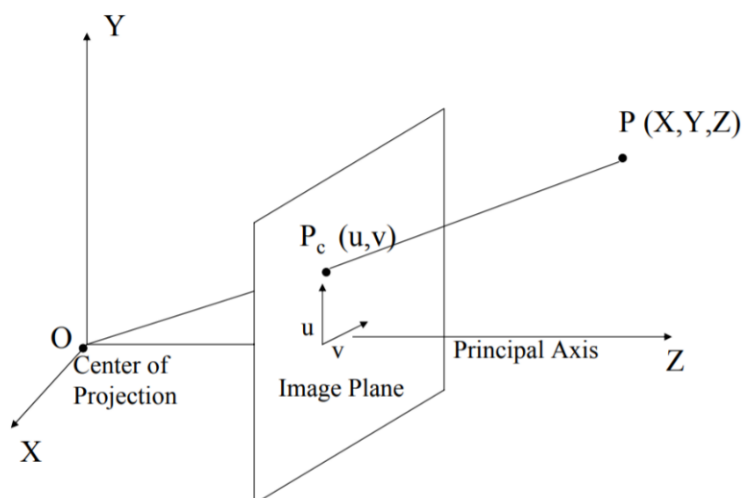


Obr. č. 2: *Príklad riadkovej kamery Basler Racer firmy atesystem (získaný z [7])*

Najpokročilejšia riadková kamera je 4CCD multispektrálna kamera, ktorá obsahuje až štyri snímače na snímanie v blízkom infračervenom pásme okolo 850 nm. Obraz z takejto kamery sa využíva na vyhodnotenie čerstvosti ovocia, zrelosti plodín na poli alebo pri inšpekcii povrchu potlačeného materiálu.

2.4 Dierkový model kamery

Dierkový model kamery (z angličtiny – pinhole camera model) popisuje matematický vzťah medzi súradnicami bodu v trojrozmernom priestore a jeho premietnutím na rovinu obrazu ideálnej dierkovej kamery, kde je clona fotoaparátu opísaná ako bod a na zaostrenie svetla sa nepoužívajú žiadne šošovky. Model nezahŕňa napríklad geometrické skreslenie alebo rozmazanie rozostrených objektov spôsobené šošovkami a clonami s konečnou veľkosťou. Nezohľadňuje tiež, že väčšina praktických kamier má iba diskkrétne obrazové súradnice. Zo znamená, že model dierkovej kamery sa dá v prvom rade použiť iba ako aproximácia mapovania z 3D scény na 2D obrázok. Dierkový model kamery je teda funkciou, ktorá modeluje skutočnú fyzickú kameru. Informácie v tejto kapitole získané z [10], [11].



Obr. č. 3: *Model kamery (získaný z [10])*

Ako môžeme na obrázku číslo 2 vidieť, 3D ortogonálny systém súradníc je v počiatku označenom O . Bod O tvorí stred premietania, je miesto, kde sa nachádza clona fotoaparátu alebo kamery. Obrazová rovina (na obrázku označená ako Image Plane) je od bodu O vo vzdialenosti značenej ako f , čo je označenie ohniskovej vzdialenosti. Keďže dierkový model kamery mapuje 3D scény na 2D obrázok, bod P je bod, ktorý sa nachádza v reálnom trojrozmernom svete na súradniciach X, Y, Z a premieta sa na 2D obrázok – na obrazovú rovinu – do bodu P_C so súradnicami u a v . Polohu tohto bodu môžeme nájsť pomocou rovníc:

$$\frac{f}{Z} = \frac{u}{X} = \frac{v}{Y} \quad (2.7)$$

Po úprave získavame súradnice bodu P_C :

$$u = \frac{fX}{Z} \quad (2.8)$$

$$v = \frac{fY}{Z} \quad (2.9)$$

Rovnice (2.7) predstavujú podobnosť trojuholníkov, ktoré môžeme nájsť na obrázku číslo 3. Získané súradnice bodu P_C môžeme zapísať do matice ako:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (2.10)$$

Avšak, pri hľadaní súradníc bodu P_C môže nastať jeden konflikt. V prípade, že sa súradnice bodu P_C nezhodujú s tým, kde by sa mala obrazová rovina pretínať s osou Z , musíme bod P_C preložiť na svoj skutočný počiatok pomocou parametrov t_u a t_v :

$$u = \frac{fX}{Z} + t_u \quad (2.11)$$

$$v = \frac{fY}{Z} + t_v \quad (2.12)$$

Pri projekcii a práci so súradnicami sú ideálne homogénne súradnice. Pre predstavu, ak použijeme euklidovské súradnice na to, aby sme určili, kde sa daný bod nachádza, použijeme n prvkov. Rozdiel medzi týmito súradnicami a homogénnymi je ten, že homogénne súradnice využívajú $(n + 1)$ prvkov. Na zápis homogénnych súradníc využívame vektor, v našom prípade je to vektor s prvkami u, v a w , ako môžeme vidieť v rovnici 2.10. Teraz môžeme povedať, že súradnice bodu P_C sú homogénne, na rozdiel od bodu P , ktorého súradnice homogénne nie sú.

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & t_u \\ 0 & f & t_v \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (2.13)$$

Keď hovoríme o obraze kamery, resp. o rozlíšení displeja monitoru, telefónu alebo fotografie, hovoríme, koľko palcov má uhlopriečka daného displeja / obrazu. Hovoríme tak preto, lebo obraz kamery je vyjadrený v jednotkách palcov. To znamená, že aj obrazová funkcia (2.13) je vyjadrená v palcoch, teda aj súradnice bodu P_C . Pri hľadaní súradníc, teda pozície P_C v pixeloch, potrebujeme poznať jeden z kľúčových parametrov kamier – údaj PPI (pixels per inch), ktorý udáva rozlíšenie kamery pomocou počtu pixelov na jeden palec. Keď teda poznáme rozlíšenie kamery m_u a m_v pixelov/palec a chceme vyjadriť pozíciu bodu P_C pomocou súradníc u a v , potrebujeme upraviť vzťah na tvar:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} m_u f & 0 & m_u t_u \\ 0 & m_v f & m_v t_v \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (2.14)$$

Tento tvar matice sme získali vynásobením jednotlivých prvkov matice pomocou rozlíšenia m_u a m_v . Týmto nám ale vzniká nová matica – matica K – ktorá v sebe obsahuje ďalšie rôzne parametre kamery. Sú to parametre kľúčové pre kalibráciu kamery a pre odstránenie skreslenia obrazu, resp. na odstránenie nedokonalostí spôsobených optickými šošovkami kamery. Túto maticu nazývame maticou vnútorných parametrov kamery, pretože udáva, aká je ohnisková vzdialenosť alebo aké je smerovanie hlavnej osi. Nový tvar matice K v sebe niekedy nesie aj parameter skosenia, značený ako s :

$$K = \begin{bmatrix} \alpha_x & s & u_o \\ 0 & \alpha_y & v_o \\ 0 & 0 & 1 \end{bmatrix} \quad (2.15)$$

Parameter skosenia sa nachádza v matici K iba vtedy, ak osi súradníc obrázka u a v nie sú voči sebe ortogonálne. Tak, ako môže nastať problém s pretínaním obrazovej roviny s osou Z , môže sa stať, že aj samotná kamera nebude mať svoj stred premietania v bode $(0,0,0)$. Tento problém sa odstraňuje pomocou rotácie a translácie na to, aby bol systém súradníc kamery zhodný so systémom, ktorý je zobrazený v obrázku číslo 3. Translácia kamery do počiatku súradníc (X, Y, Z) je značená ako $T = (T_x, T_y, T_z)$ (2.16) a rotácia, s ktorou pomocou aplikácie dokážeme, aby sa os Z zhodovala s hlavnou osou, zapisujeme pomocou rotačnej matice, značenej ako R , o rozmeroch (3×3) . Výsledkom prvej translácie a následnej rotácie je matica vonkajších parametrov kamery – matica E – ktorá v sebe obsahuje údaje o rotácii a posunutí kamerového systému:

$$E = \begin{bmatrix} R & -RT \\ 0 & 1 \end{bmatrix} \quad (2.17)$$

Na vyjadrenie konečnej transformácie kamery musíme vyjadriť bod P v homogénnych súradniciach. Vďaka tomuto kroku bude bod P_c projekciou bodu P .

$$P_c = K \begin{bmatrix} R & -RT \\ 0 & 1 \end{bmatrix} P = KR \begin{bmatrix} 1 & -T \\ 0 & 1 \end{bmatrix} = KR[I \quad -T]P = CP \quad (2.18)$$

2.5 Kalibrácia videokamery

Dnešné kamery spôsobujú v snímkach zakrivenie obrazu a veľké skreslenie. Jedná sa o radiálne skreslenie a tangenciálne skreslenie. Toto optické skreslenie je spôsobené šošovkou (šošovkami) kamery alebo fotoaparátu. Ak by sme fotili priamky alebo rovné tvary, v dôsledku radiálneho skreslenia sa tieto tvary budú javiť zakrivené (vypuklé) a toto zakrivenie s väčšou vzdialenosťou od stredu obrazu rastie. Skreslenie patrí medzi tzv. monochromatické aberácie a delí sa podľa spôsobu vzniku na tri druhy: poduškové skreslenie, súdkovité skreslenie a zvlnenie. Poduškové skreslenie vzniká pri fotení s dlhými objektívmi a súdkovité skreslenie pri fotení so širokouhlými objektívmi. Kalibráciou kamery sa snažíme toto skreslenie odstrániť pomocou rovníc pre radiálne a tangenciálne skreslenie. Informácie v tejto získané z [13], [9], [2].

2.5.1 Radiálne skreslenie

Radiálne skreslenie vzniká poduškovým alebo súdkovitým spôsobom a je spôsobené šošovkou v kamere. Pri kalibrácii kamery sa najčastejšie uvažuje len s radiálnym skreslením šošovky, pretože sa dá matematicky dobre popísať a kalibráciou kamery aj eliminovať z výsledného obrazu. Skreslený obraz sa upravuje tak, že body obrazu sú podrobené transformácií, ktorá závisí len od vzdialenosti od hlavného bodu,

ktorý skresleniu nepodlieha. Čím viac je transformovaný bod vzdialený od hlavného bodu, tým viac je jeho poloha skreslená. Radiálne skreslenie sa rieši nasledovne:

$$x_{opravené} = x(1 + k_1r^2 + k_2r^4 + k_3r^6) \quad (2.19)$$

$$y_{opravené} = y(1 + k_1r^2 + k_2r^4 + k_3r^6) \quad (2.20)$$

2.5.2 Tangenciálne skreslenie

Ku tangenciálnemu skresleniu dochádza preto, lebo šošovka snímajúca obraz nie je dokonale zarovnaná rovnobežne s rovinou zobrazovania a vo výsledku môžu niektoré časti obrázka vyzeráť bližšie, ako by mali. Tangenciálne skreslenie sa rieši takto:

$$x_{opravené} = x + (2p_1xy + p_2(r^2 + 2x^2)) \quad (2.21)$$

$$y_{opravené} = y + (p_1(r^2 + 2y^2) + 2p_2xy) \quad (2.22)$$

Na to, aby sme mohli skreslenie obrazu odstrániť je potrebné nájsť celkovo päť koeficientov skreslenia – koeficienty k_1, k_2, k_3, p_1 a p_2 . V rovniciach radiálneho a tangenciálneho skreslenia sa objavuje aj premenná r , ktorá tvorí vzdialenosť od stredu obrázka. Vypočítame ju ako:

$$r = \sqrt{x^2 + y^2} \quad (2.23)$$

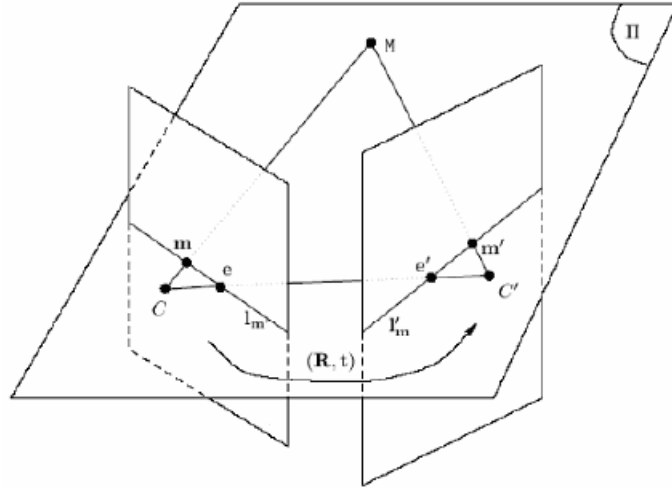
Na úspešné odstránenie skreslenia kamery sú potrebné aj ďalšie informácie, ako napríklad vnútorné a vonkajšie parametre kamery. Vnútorné parametre sú špecifické pre kameru, napríklad ohnisková vzdialenosť (f_x, f_y), optické centrá (c_x, c_y) atď. Vnútorné parametre kamery sa uvádzajú v matici kamery:

$$matica\ kamery = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 0 \end{bmatrix} \quad (2.24)$$

Vonkajšie parametre zodpovedajú vektorom rotácie a translácie, ktoré prekladajú súradnice 3D bodu do súradnicového systému. Matica kamery je kalibračná matica, ktorú sme mohli bližšie spoznať už vo vzťahu (2.15).

2.5 Epipolárna geometria

Na to, aby človek objekty vo viacerých rozmeroch, potrebuje dve oči. Vďaka tomu, že sú oči na dvoch rôznych miestach lebky, dokáže sledovať objekt z viacerých rozmerov a dokáže vnímať jeho rozmery a lepšie sa orientovať. Keď pozeráme film v 3D, potrebujeme špeciálne okuliare s filtrami na to, aby sme videli dva rozdielne obrazy premietané naraz na jedno plátno. Každý z týchto dvoch obrazov bol natočený rozdielnymi kamerami uloženými blízko pri sebe v jednej rovine. Tieto kamery sú tak uložené preto, aby ľudské oko vnímalo tento obraz čo najprirodzenejšie. Ďalším parametrom prirodzenosti pre ľudský zrak je aj vzdialenosť medzi kamerami – čím viac sa vzdialenosť medzi kamerami približuje vzdialenosti medzi očami, tým príjemnejšie je sledovanie výsledného obrazu. Snímanie obrazu bližšie opisuje epipolárna geometria, ktorá vypočítava 3D súradnice objektu alebo scény z dvoch jej 2D obrázkov. Na úspešné určenie 3D súradníc využíva táto geometria vnútorné parametre kamery a vzájomnú polohu kamier. Medzi vnútorné parametre radíme napríklad ohniskovú vzdialenosť kamier, pozíciu riadiacich bodov, deformáciu, zakrivenie, atď.. Za vonkajšie parametre považujeme posunutie alebo natočenie snímok voči sebe. Informácie v tejto kapitole získané z [1], [16].



Obr. č. 4: Zachytenie trojrozmerného bodu M pomocou dvoch kamier (získaný z [1])

2.5.1 Parametre kamery

Dierkový model kamery nám bližšie opísal, ako funguje kamera. Zaviedli sme si bod O tvoriaci stred premietania. Bolo to miesto, kde sa nachádzala clona fotoaparátu alebo kamery. Ďalej sme si bližšie špecifikovali bod P , ktorý sa nachádzal v reálnom trojrozmernom svete na súradniciach X, Y, Z a premietal sa na 2D obrázok – na obrazovú rovinu – do bodu P_C so súradnicami u a v . Na obrázku č. 4 sú stredmi premietania body C a C' a bod M je reálny trojrozmerný bod. Bod m sa nachádza v obrazovej rovine a je to projekcia bodu M . Body v rovinách označené ako e a e' nazývame epipóly, ktoré označujú miesta, na ktorých je možné vidieť druhú kameru. Ďalej tu môžeme vidieť aj epipolárne priamky l_m a l'_m , na ktorých sa nachádza bod m a m' . Ak sa bod m nachádza v prvej rovine na priamke l_m , tak sa na priamke l'_m môže nachádzať na ktoromkoľvek mieste. Rovnaký princíp platí aj pre bod m' . Tieto priamky tvoria epipolárnu rovinu π , množinu, ktorú tvoria body všetkých potenciálnych pozícií bodu M .

2.5.2 Fundamentálna matica

Na využitie poznatkov epipolárnej geometrie slúži fundamentálna matica, ktorá pomocou algebry udáva informácie o polohe dvoch kamier a ich vnútorné parametre. Udáva vzťah medzi bodmi m a m' v obrazových rovinách. Medzi fundamentálnou maticou F , týmito bodmi platí vzťah a epipolárnou priamkou l'_m platí vzťah:

$$m^T F m' = 0 \quad (2.25)$$

$$l'_m = F p \quad (2.26)$$

V jednom zo spôsobov, pomocou ktorého získame fundamentálnu maticu F využívame esenciálnu maticu E , ktorú sme si uviedli vo vzťahu (2.17). Táto matica obsahuje iba vonkajšie parametre kamery a nie vnútorné - obsahuje údaje o rotácii a posunutí kamerového systému a vzájomnú pozíciu kamier.

$$F = K_2^{-T} R S K_1^{-1} \quad (2.27)$$

K_1 a K_2 sú kalibračné matice použitých kamier. Súčinom matíc R a S dostávame esenciálnu maticu E . Výslednú fundamentálnu maticu môžeme teda zapísať ako:

$$F = K_2^{-T} E K_1^{-1} \quad (2.28)$$

Pomocou fundamentálnej matice F , ktorá je maticou druhého stupňa so siedmimi parametrami – dvomi pre každú epipólu a štyrmi pre homografiu spájajúcu obrazové

roviny - môžeme vyjadriť vzťah medzi korešpondenčnými bodmi, ktoré budú v pixelových súradniciach.

$$l_m'^T F l_m = 0 \quad (2.29)$$

3 Realizácia

3.1 Knižnica OpenCV

OpenCV je open-source knižnica dostupná pre rôzne programovacie jazyky na programovanie v oblasti počítačového videnia. Je vyvíjaná v jazyku C a C++, no je možné ju využiť aj v jazyku Python. Okrem jazyka C, C++ alebo Python je možné využitie aj v Jave alebo Matlabe. Je priamo určená na spracovanie obrazu a poskytuje množstvo funkcií, vďaka ktorým je používateľ schopný vypočítať napríklad kalibračnú maticu alebo fundamentálnu maticu. Okrem toho obsahuje ďalšie rôzne funkcie, ako napríklad geometrickú transformáciu, filtre, detekciu objektov v obraze a je vyvíjaná na počítačové operačné systémy, ako sú Windows, Linux a Mac OS, ale aj mobilné operačné systémy, teda Android a iOS.

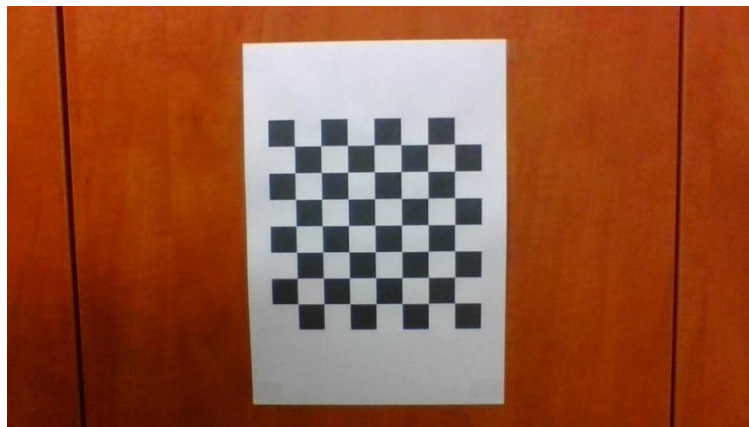
3.2 Programovací jazyk Python

Python je interpretovaný, interaktívny programovací jazyk, vyvíjaný ako open-source projekt, podporujúci objektovo orientované, štruktúrované aj funkcionálne programovanie. Tak, ako je OpenCV napísané v jazyku C a C++, rovnako aj ďalšie moduly jazyka Python môžu byť jednoducho napísané v C alebo C++. Jeho štandardná knižnica je často využívaná pri tvorbe aplikácií, ktoré pracujú napríklad s internetom,

grafickým prostredím, relačnými databázami alebo regulárnymi výrazmi. Jeho ďalšou výhodou je zobrazovanie informácií do terminálu, vďaka čomu môže programátor, okamžite vidieť výsledok.

3.3 Kalibrácia kamery

Pred začiatkom kalibrácie kamery je potrebné nainštalovať programovací jazyk Python verzie 3.8 (aktuálna najnovšia verzia) a k tomu nainštalovať knižnicu OpenCV. Inštalačné súbory sú ľahko dostupné z originálnych zdrojov vydavateľa jazyka Python. Na kalibráciu kamery je potrebný aj vytlačený obrázok šachovnice, ktorý môže byť nalepený na stenu alebo na nejakú podložku. Je dôležité, aby šachovnica bola v jednej rovine a nebola zdeformovaná, kvôli presnosti kalibrácie. Ďalším krokom je vyhotovenie minimálne desiatich snímok šachovnice z rôznych uhlov. Toto zaručuje lepšiu presnosť pri počítaní skreslenia obrazu a jeho následného odstránenia. V kalibračnom algoritme sú tieto snímky načítané vo formáte *.jpg* z priečinka, v ktorom sa nachádza aj zdrojový kód pre kalibráciu. Načítanie prebehne príkazom *glob.glob('*.*jpg')*, ktorý je cez príkaz *import* importovaný na začiatku algoritmu spolu s ďalšími modulmi – modulmi *cv2* ako *cv* a *numpy* ako *np*, vid' priložený kód. Okrem týchto obrázkov sú na začiatku algoritmu definované aj kritériá maximálneho počtu iterácií a epsilon, ktoré sú kľúčové pri hľadaní jednotlivých rohov šachovnice. Informácie získané z [2].



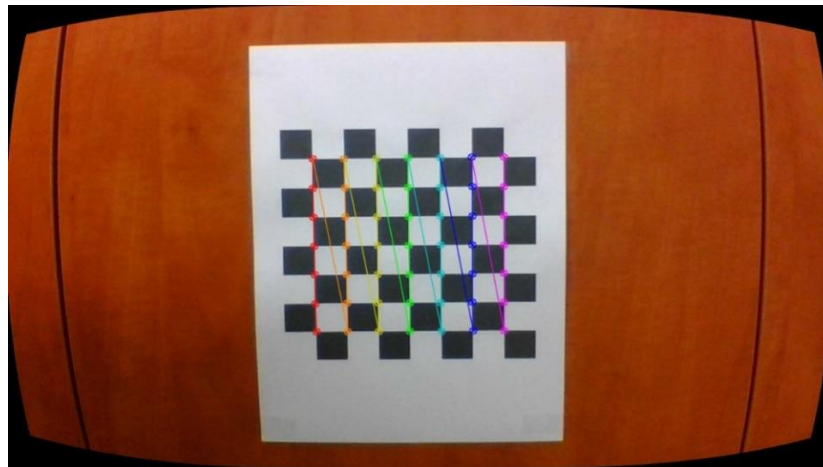
Obr. č. 5: Jeden z desiatich odfotených obrázkov šachovnice pred kalibráciou

Po týchto úvodných krokoch nastáva samotná kalibrácia. Načítané obrázky sú prevedené z farebného formátu na čiernobiely pomocou funkcie *cvtColor*. Je to nutné z dôvodu, že farebné obrázky obsahujú šum a neostrosti, ktoré sa prevodom na čiernobiely obrázok aspoň čiastočne eliminujú. Na odfotenej šachovnici z rôznych strán sú nájdené rohy rôznych štvorčekov cez príkaz *findChessboardCorners*. Šachovnica použitá na kalibráciu má rozmery 7×7 štvorčeka, čo je tiež jeden z potrebných parametrov hľadanie rohov. Na jednotlivých rohoch štvorčekov sa objavujú bodky, ktoré sú medzi sebou jednotlivo poprepávané a presnosť ich lokácie je docielená funkciou *cornerSubPix*, teda, ak sa rohy našli. Nájdené rohy sa aplikujú na farebný obraz a výsledok je takýto:



Obr. č. 6: Nájdené rohy štvorčekov šachovnice rozmerov 7×7

Najdôležitejšia a najhlavnejšia je funkcia *calibrateCamera*, ktorá kalibruje obraz kamery. Vracia kalibračnú maticu (2.15) v premennej *mtx* a koeficienty zakrivenia v premennej *dist*. Pracuje s nájdenými rohmi šachovnice a s priamkami, ktoré sú medzi nimi. Vypočíta, ako sú úseky šachovnice nerovné a v akých odchýlkach od seba vzdialené. Podľa tejto zmeranej oblasti 7×7 štvorčeka sa nakoniec vytvorí ideálne riešenie, ktoré odstráni zakrivenie pri ďalšom používaní kamery. Kalibračnú maticu a koeficienty zakrivenia potrebuje *getOptimalNewCameraMatrix* na upravenie a výsledné skalibrovanie získame funkciou *undistort*.



Obr. č. 7: Výsledný skalibrovaný obraz

Výsledkom kalibrácie nie je iba obrázok číslo 7, na ktorom môžeme vidieť, ako kalibrovanie vyrovnať priamky medzi jednotlivými rohmi štvorčekov, ale aj kalibračnú maticu (3.1), ktorá je využitá pri ďalšom spracovaní obrazu. Okrem tejto matice sú k dispozícii aj koeficienty skreslenia obrazu (3.2).

$$mtx = \begin{bmatrix} 9,9629 \cdot 10^2 & 0 & 6,5238 \cdot 10^2 \\ 0 & 9,9810 \cdot 10^2 & 3,4092 \cdot 10^2 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

$$dist = [-1,9455 \cdot 10^{-1} \quad 7,5619 \cdot 10^{-1} \quad -2,8084 \cdot 10^{-3} \quad 3,6296 \cdot 10^{-3} \quad 1,253] \quad (3.2)$$

3.4 Simulácia riadkovej kamery

Na meranie riadkovou kamerou stačí aj bežná kamera s rozlíšením napríklad 1280×720 pixelov. Programovací jazyk Python a knižnica OpenCV si s obrázkami vedia poradiť ako s maticami. Pri tvorbe obrazu pomocou riadkových kamier sa berie z obrázka iba jeden riadok pixelov, teda jeden riadok matice. Na zloženie väčšieho snímku je potrebné pospájať dokopy niekoľko takýchto riadkov.

Pre začiatok je potrebné určiť, ako má riadková kamera fungovať a ako bude prebiehať získanie niekoľkých riadkov z rôznych snímok. Vhodná je napríklad funkcia *VideoCapture*, ktorá neustále sníma z kamery obraz vo frekvencii 24 fps (viď kapitola 2.3). Tento obraz je ale potrebné skalibrovať použitím kalibračnej matice (3.1) a koeficientov skreslenia obrazu (3.2) na odstránenie skreslenia. Na to slúžia funkcie *getOptimalNewCameraMatrix* a *undistort*. Na odstránenie šumu a neostrości sú 2 metódy. Prvá metóda je metóda, ktorá bola použitá pri kalibrácii kamery – prevod farebného obrazu na čiernobiely. Druhá metóda je taká, že po kalibrácii skresleného obrazu je získaný nový obraz s odstráneným skreslením, čo je najlepšie vidieť v jeho strede. Na vytvorenie jedného riadku si zobraných päť riadkov zo stredu pomocou funkcie *frame*, ktorá oreže skalibrovaný obraz s rozlíšením 1280×720 pixelov na 1280×5 pixelov. Päť pixelov, teda riadkov matice, sa spojí do jedného.

Na spojenie týchto riadkov do ucelenej snímky je potrebná ďalšia matica, ktorá môže byť zadefinovaná napríklad tak, že je do nej nahraný obrázok so šachovnicou (obr. č. 5). Tým je získané potrebné rozlíšenie pre nový obraz. Riadky tejto matice sú označené indexom *i* a jeho postupným zvyšovaním prebieha vkladanie výsledného riadku do novej, ucelenej snímky. Pretože zadefinovaná matica má rozmery 1280×720 pixelov, musí kamera získať až 720 snímok, ktoré sú spracované podľa krokov spomenutých vyššie. Kľúčovými funkciami v tomto procese sú príkazy *frame* a *resize*.



Obr. č. 8: Peračník odfotený pomocou riadkovej kamery

Ďalšie pokusy o fotografie pomocou riadkových kamier sú v kapitole *Prílohy*, kde sa nachádzajú dva rôzne pohľady na meraný objekt. Pre porovnanie kvality fotografie a merania sú v kapitole priložené aj vzorové fotografie, ktoré boli odfotené plošnou kamerou. Kvôli rozmerom sú tieto objekty orezané. Obrázky č. 9. až č. 12 boli odfotené z dvoch rôznych polôh objektu a obrázky č. 13. až 15. boli odfotené z jednej polohy. Rôzne pohľady tu vznikli softwarovou úpravou, kde sme si zo vstupného obrazu zobrali rôzne riadky a tie spojili.

3.4.1 Zhodnotenie merania pomocou simulácie riadkovej kamery

Snímky z takejto simulácie však spôsobujú oproti skutočnej riadkovej kamere veľké nepresnosti. Meranie je komplikované, nemusí byť zaistený konštantný posun

v jednom smere, môže vzniknúť výchylka v rýchlosti posunu, resp. sa môžeme vychýliť do strán. Príklad takéhoto pracoviska je na obrázku č. 9. Podložka, na ktorej bol položený objekt, bola zaistená o roh stola a posúvaná smerom ku webkamere. Telefón slúžil ako zdroj svetla. Výsledky z merania týmto spôsobom sú však príliš zdeformované, strácal sa skutočný tvar snímaného objektu. Počas merania sa menila rýchlosť snímania kamery, čo pravdepodobne spôsobovali slabé parametre kamery v spojení so slabými svetelnými podmienkami. Preto bola pre ďalšie meranie použitá skutočná riadková kamera s posuvným stolčekom umiestneným priamo pod kamerou.

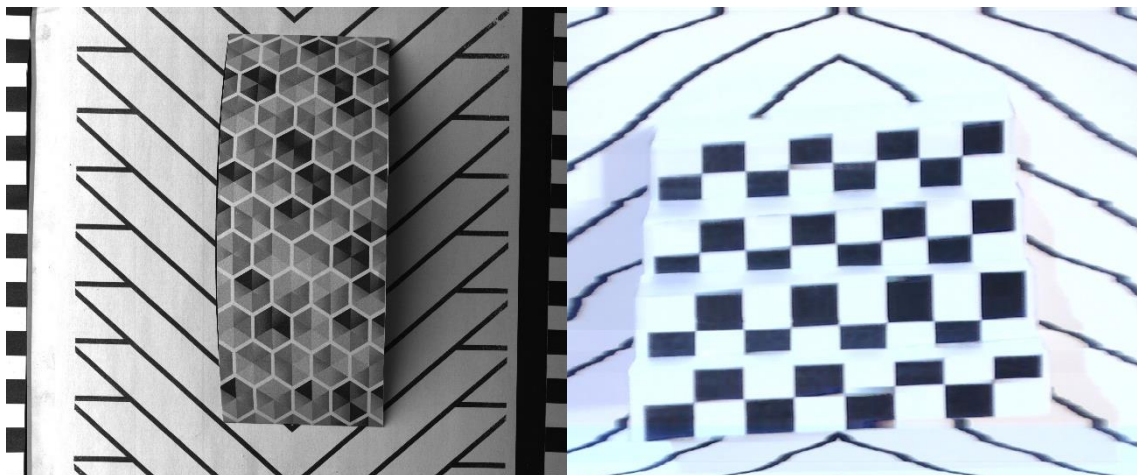


Obr. č. 9: Meracie prostredie použité pri simulácii riadkovej kamery

3.5 Meranie pomocou reálnej riadkovej kamery

3.5.1 Porovnanie kamery so simuláciou

Meranie pomocou riadkovej kamery dáva snímky v lepšej kvalite. Vzniká tu však problém s pomerom strán snímok, ktorý je iný, ako ten skutočný, s ktorým by používateľ pracoval. Spôsobuje to aj výhodu, ktorou je vyššie rozlíšenie a to pri meraní zlepšuje presnosť pri aplikovaní algoritmov. Pri obsluhovaní riadkovej kamery môžu vzniknúť nepresnosti v podobe rôznych geometrických uhlov merania. Kamera všeobecne nemusí byť priamo nad meracím objektom a môže byť vychýlená do strany, môže byť pootočená do ľavej alebo pravej strany, poprípade riadkový čip nie je vodorovne s meracou



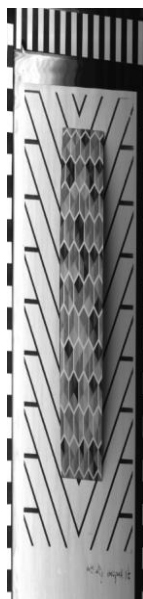
podložkou. Na všetky tieto uhly a nepresnosti je potrebné počas aplikovania algoritmov myslieť a tvoriť korekcie obrazu, ktoré by tieto nedostatky odstránili. Matematický model merania je pre minimálne odchýlky veľmi dôležitý.

Obr. č. 10: *Porovnanie snímok zo skutočnej riadkovej kamery (vľavo) a simulovanej (vpravo)*

3.5.2 Výpočet správneho pomeru strán snímok (aspect ratio)

Snímka z riadkovej kamery na obrázku č. 10 vľavo je snímka, ktorá má upravený pomer strán (aspect ratio). Vytvorený algoritmus dokáže upraviť rozlíšenie 4600 pixelov na šírku a 19 500 na výšku tak, aby boli zachované skutočné rozmery meracieho prostredia. Po krajoch podložky sa nachádza šachovnica s rozmerom obdĺžnika jeden centimeter. Táto šachovnica sa nachádza na všetkých štyroch stranách meracieho prostredia, čo umožňuje presne určiť, aké sú skutočné rozmery a aké skreslenie vo výške snímky vzniká. Aj napriek tomu, že bola snímka orezaná na šírku 4600 pixelov, je možné vypočítať, koľko má jeden centimeter pixelov a porovnať ho s počtom pixelov centimetra, ktorý je na výšku natiahnutý. Na výpočet skutočného pomeru strán v pixeloch postačí jednoduchá matematika. Výsledná snímka má rozlíšenie 4600x5434 pixelov. Takáto snímka dáva lepšiu predstavu o tom, ako meracie prostredie vyzerá a objekt v tvare paraboly je ľahko merateľný. Známe sú však aj jeho rozmery - jeho šírka je 7 centimetrov v osi X, 16 centimetrov v osi Y a v strede je vo výške maximálne 2,5 centimetra. S rastúcou výškou sa viac rozširuje do strán, pretože zaberá väčší priestor v nami pozorovanom uhle. Zmeraním šírky objektu v každom riadku, v ktorom sa nachádza, je kľúčový krok pri určovaní jeho výšky. Zároveň je možné vykonať korekciu objektu tak, aby sa s rastúcou výškou nerozširoval a aby si zachoval svoje rozmery.

Algoritmus, ktorý mení pomer strán, najprv načíta pôvodnú snímku a pomocou príkazu shape zistí, aké sú jeho rozmery. Tieto hodnoty uloží do svojich samostatných premenných a do terminálu tieto rozmery vypíše. Okrem hlavnej snímky načíta aj orezané snímky, ktoré nesú informácie o šírke objektu v miestach, kde by malo byť skreslenie výškou minimálne. Na zistenie týchto rozmerov je možné použiť algoritmy, ktoré poskytujú údaje o polohe pixelov, resp. medzi nimi počítajú vzdialenosť. Na takúto kalibráciu je vhodné použiť aj údaje o šachovnici po krajoch. Toto však môže byť pri aktuálnych snímkach nepresné, pretože kamera bola naklonená do strany a kraje snímky by nemuseli dať presnú informáciu o tom, aké sú reálne rozmery. Stred obrázka niesol v niektorých snímkach malé skreslenie s malou výchyľkou. Cieľom takéhoto postupu bolo aj osvojenie si práce s načítavaním a spracovaním viacerých snímok.

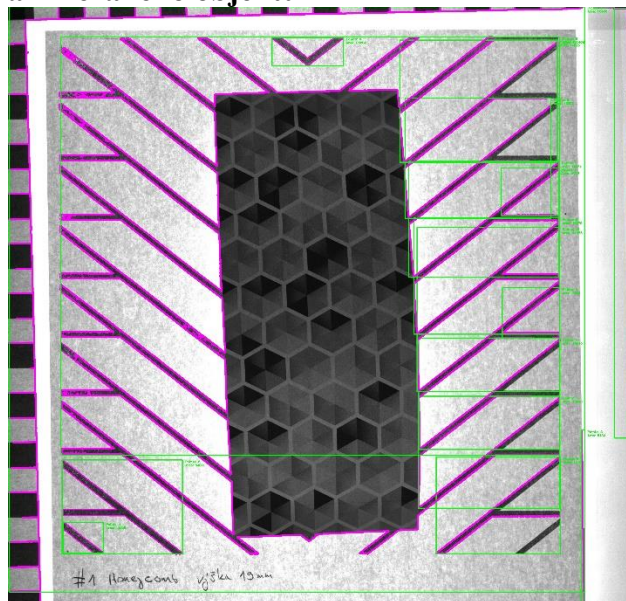


Obr. č. 11: *Originálna snímka z riadkovej kamery v plnom rozlíšení*

Na obrázku č. 11 je vidno viacero problémov, ktoré je treba odstrániť. Prvý problém je vychýlenie kamery do strany, čo spôsobuje, že podložka nie je vodorovne a je mierne vyrotovaná. Rovnaký problém sa nachádza aj na samotnom meranom objekte, na ktorom vidno, že horná hrana nie je vodorovne so spodnou. Toto spôsobuje vznik asymetrie v podobe rôznych výšok objektu v jednom riadku, teda v rovine, ktorá predstavuje pohľad z kamery. Na odstránenie týchto problémov stačia základné znalosti geometrie.

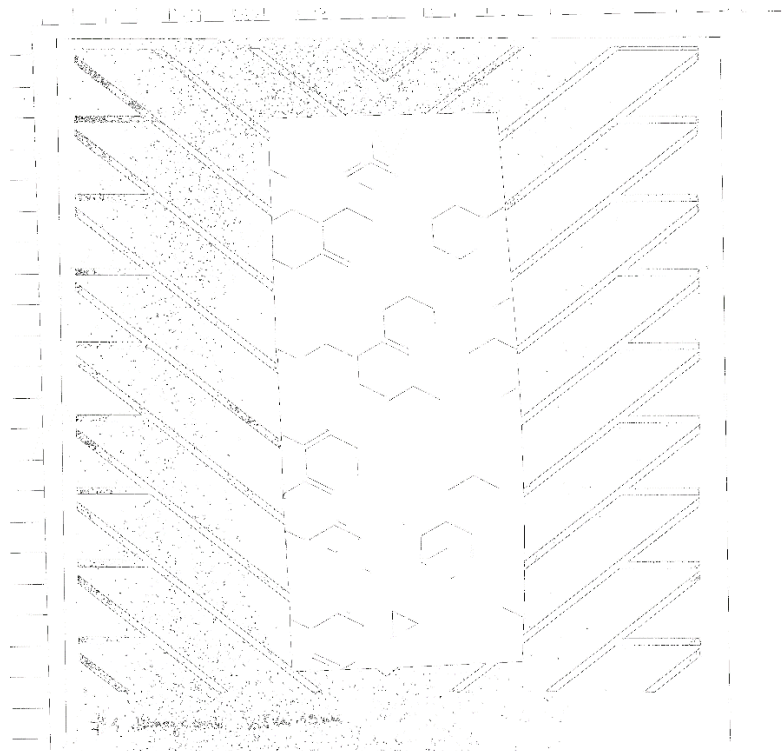
Pri návrhu algoritmu, ktorý by automaticky meral rozmery objektu, bolo testovaných viacero filtrov na detekciu hrán. Meraný objekt v tvare paraboly nesie vo svojom vzore veľké množstvo hrán, ktoré nie je problém zdetekovať. Ako príklad je uvedený obrázok č. 12 s Cannyho filtrom. Okrem tohto filtra existujú aj ďalšie, napríklad Moravcov filter detekcie hrán alebo Harrisov filter detekcie hrán.

3.5.3 Detekcia hrán meraného objektu



Obr. č. 12: Detekcia hrán

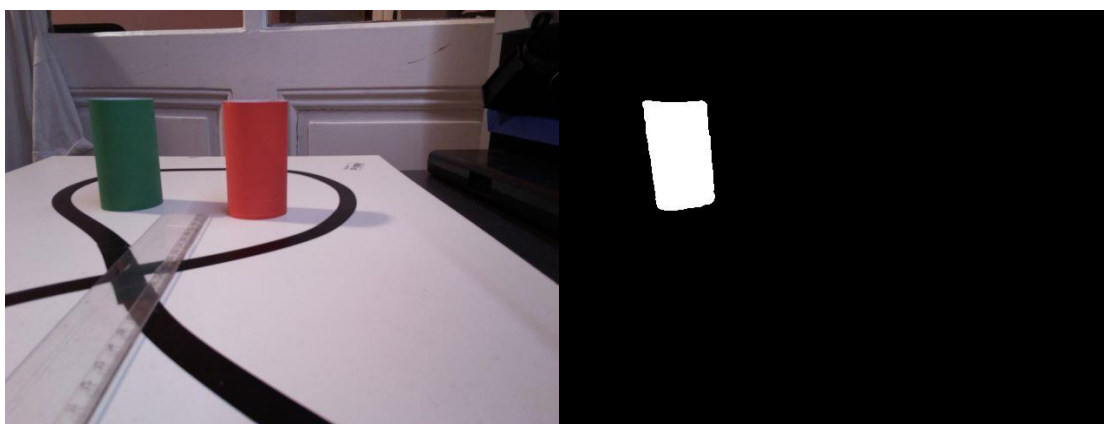
Použitá verzia algoritmu je výpočtovo náročnejšia, pretože pracuje s plným nezmeneným rozlíšením snímky na dosiahnutie lepšej presnosti. Pri hľadaní kontúr je potrebné meniť viacero parametrov, takzvaný *threshold 1*, *threshold 2* a *area*, ktorá udáva počet pixelov, ktoré by sa mali v oblasti nachádzať, aby mohla byť označená za kontúru. Na obrázku č. 12 je viacero údajov. Medzi tie patria napríklad ružové čiary, ktoré predstavujú hrany a zelené obdĺžniky, ktoré vyznačujú kontúry. Tento algoritmus pomáha určiť, kde sa nachádza meraný objekt a na základe polohy ružových pixelov počíta, kde sa nachádzajú hrany objektu. Testovaných bolo viacero grafických vzorov pripevnených na parabole. Medzi tie patrí napríklad tmavý Honeycomb z obrázku č. 12, svetlejší Honeycomb z obrázkov č. 10 vľavo a z obrázku č. 11. K dispozícii bol aj štvorcový – šachovnicový – s malými štvorčekmi, ďalší mal štvorčeky väčšie a ďalší menšie, pootočené o 45°. Detekcia prebiehala pri každom vzore inak, pretože niektoré vzory boli tmavšie a niektoré svetlejšie. Medzi podložkou a meraným objektom preto nebol veľký rozdiel a niektoré ružové pixely sa začali vykresľovať aj v samotnom meranom objekte, čo bolo nechcené. Zmenou *thresholdov* je možné filtrovať a upravovať detekciu tak, aby sa žiadne pixely do objektu nedostali. Na zmenu parametrov v reálnom čase slúži trackbar, teda posuvný ukazovateľ na osi s hodnotami od 0 do 255. Na základe pozície ukazovateľa sú získavané a volené nové hodnoty prahovania – *thresholdu*. Na detekciu kontúr je použitý Cannyho filter, ktorého výsledok sa nachádza na obrázku č. 13. Medzi jeho výhody patrí minimálna chybovosť, lokalizácia a jednoznačná odozva. V algoritmoch, ktoré tento filter využívajú, sa často aplikuje Gaussov filter na to, aby sa odstránil šum. Tento krok obrázkov rozmazá a kvôli presnosti nebol aplikovaný. Odstránenie šumu zaručuje splnenie prvej podmienky – čo najmenší počet chýb pri detekcii hrán. Snímky s vysokým rozlíšením majú výhodu a pre dosiahnutie presnejších výsledkov netreba rozmazanie využiť. Na záver algoritmu je zmenený pomer strán obrázku tak, aby rozmery odpovedali realite. Toto tiež eliminuje prípadné výchyľky a nepresnosti vzniknuté pri aplikovaní Cannyho filtra.



Obr. č. 13: *Cannyho filter na detekciu hrán*

3.5.4 Detekcia objektu pomocou plošnej kamery

Ako príklad detekcie objektu je uvedené meranie v obrázku č. 14, ktoré využívalo plošnú kameru. Snímky, ktoré boli kamerou vyhotovené, mali rozmery 640x480 pixelov a cieľom bolo meranie výšky a šírky objektu v obraze. Využívalo sa pri tom maskovanie, ktoré vznikalo rozdelením kanálov obrázka na tri rôzne zložky – červenú, zelenú a modrú. Ako prvý krok po získaní obrazu bol aplikovaný Gaussov filter, ktorý obraz rozmazal za účelom menšej chybovosti. V tomto meraní bolo dôležité osvetlenie, ale aj samotný farebný model. RGB farebný model dosahoval horšie výsledky ako HSV farebný model, ktorý lepšie detekoval zelenú farbu.



Obr. č. 14: *Príklad detekcie objektu*

Na obrázku č. 14 vľavo sa nachádza zelený objekt, ktorého maska sa nachádza na obrázku č. 14 vpravo. Takúto presnosť umožňuje HSV farebný model, ktorý dokáže zelenú farbu bez problému zdetekovať, na rozdiel od RGB farebného modelu, ktorý zelenú farbu nezdetegoval.

3.6 Overovanie algoritmov na tvorbu 3D modelu

3.6.1 Porovnávacie algoritmy

Široké spektrum algoritmov tvoria porovnávacie algoritmy, ktoré hľadajú spoločné body na rôznych obrázkoch. Túto úlohu vykonávajú napríklad algoritmy *Sift*, *Surf*, *Brisk* alebo *Akaze*. Funkcia *Sift* už nie je v nových vydaniach OpenCV (posledná verzia, v ktorej bolo *Sift* prítomné, bola 2.4.6, aktuálna verzia OpenCV v čase písania je 4.3.0) podporovaná a počas testovania porovnávania obrázkov dochádzalo k rôznym problémom s kompatibilitou knižnice OpenCV. V ďalšom postupe bola preto pre testovanie zvolená funkcia *Akaze*. Je to algoritmus, ktorý porovnáva dva rôzne obrázky a hľadá na nich spoločné body. Nakoniec ich prepojí epipolárnymi priamkami, ktoré je možné medzi dvoma obrázkami vykresliť. Na to, aby to dokázal, potrebuje maticu homografie. Tá sa dá získať vybraním štyroch spoločných bodov z dvoch obrázkov a vložením ich súradníc X a Y do algoritmu. Pomocou funkcie *findHomography* získaná matica, ktorú je ďalej možné využiť a hľadať spoločné body dvoch obrázkov. Nájdene body nazývané *keypoints*, teda kľúčové, získané pomocou funkcie *detectAndCompute* z načítaných obrázkov. Homografia je transformácia mapujúca body v jednom obrázku s korešpondujúcimi bodmi v druhom obrázku. Ak tieto body ležia v skutočnom svete v rovnakej rovine, je tieto snímky možné zarovnať tak, aby sa nachádzali v rovnakej pozícii a v rovnakom uhle natočenia – v jednej rovine.

$$H = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \quad (3.3)$$

$$H = \begin{bmatrix} 1,0000 & -4,5612 \cdot 10^{-7} & -2,0010 \cdot 10^1 \\ 7,6864 \cdot 10^{-4} & 9,9997 \cdot 10^{-1} & -2,2131 \cdot 10^1 \\ 2,3979 \cdot 10^{-9} & -2,1036 \cdot 10^{-10} & 1,0000 \end{bmatrix} \quad (3.4)$$

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = H \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} \quad (3.5)$$

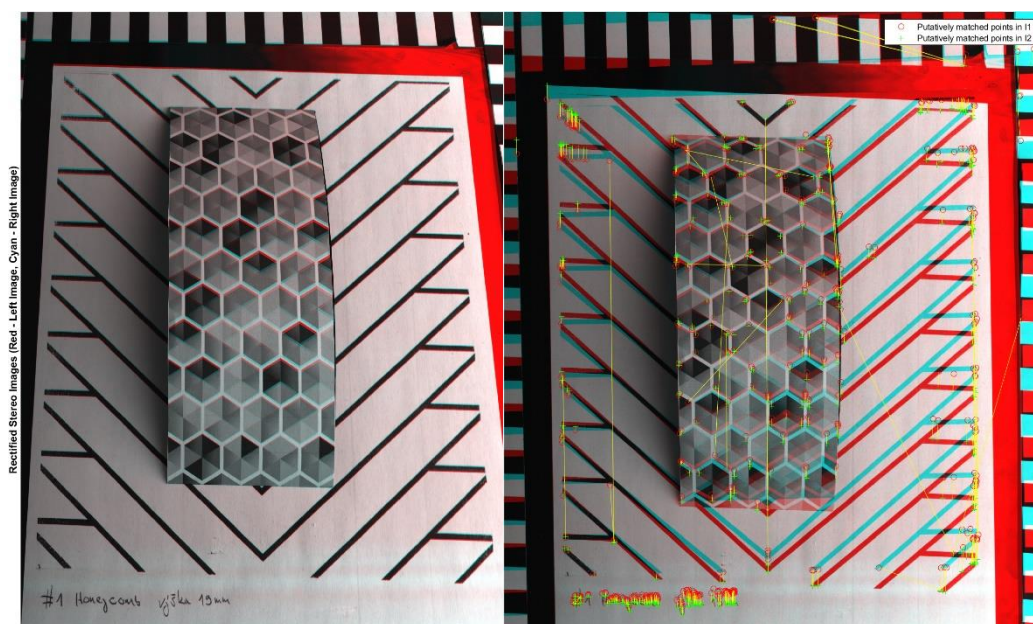
Matica homografie (3.4) je výstupná matica z funkcie *findHomography*, ktorú algoritmus využíva pri hľadaní spoločných bodov z dvoch obrázkov. Maticu homografie je potrebné dosadiť do rovnice 3.5. Ak sú body x_1 a y_1 v jednom obrázku a body x_2 a y_2 v druhom, matica homografie ich zmapuje.



Obr. č. 15: Výsledok hľadania spoločných bodov

Pri aplikovaní algoritmov 3D rekonštrukcie je vhodné pracovať s parametrami, resp. s dátami, ktoré poskytujú algoritmy na kalibráciu kamery. V kapitole 2.5 a 3.3 je bližšie uvedené, čo je kalibrácia kamery a ako sa aplikuje. Vo výstupe kalibračnej funkcie je maticu kamery, koeficienty skreslenia, vektory rotácie a translácie. Pri aplikovaní algoritmov na snímky z plošných kamier nenastáva žiaden problém. Skreslenie môže nastávať v smere X a Y, pri riadkovej kamere môže nastávať iba v smere X, teda v riadku. Ak by bol pomocou riadkovej kamery odfotený šachovnicový vzor, zmenil sa pomer strán snímky a aplikoval algoritmus kalibrácie kamery, výsledky kalibrácie by boli iba odhad, ktorý by pri spracovaní obrazu nemusel fungovať. Existujú funkcie, ktoré niektoré parametre vedia odhadnúť, no tieto hodnoty by neboli presné. Príkladom takejto funkcie je *estimateFundamentalMatrix*, ktorá dokáže odhadnúť fundamentálnu maticu z kapitoly 2.5.2.

3.6.2 Rektifikácia obrazu



Obr. č. 16: Rektifikácia snímok (vľavo) a nájdenie spoločných bodov (vpravo)

Fundamentálnu maticu býva používaná na zosúhlasenie dvoch snímok, aby ležali v jednej rovine. Tento proces sa nazýva rektifikácia obrazu. Pri meraní je potrebné vedieť, ako sa originálne snímky pri spracovaní menia a čo to znamená pre meranie objektu. Následne by sa zistili posuny rovnakých detailov na parabole – meranom objekte, ktoré odpovedajú výške.

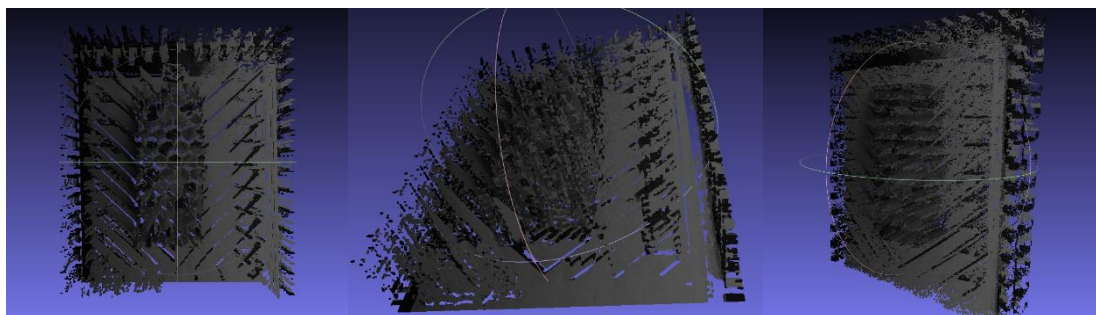
$$F = \begin{bmatrix} 1,1070 \cdot 10^{-7} & -4,6126 \cdot 10^{-6} & 0,3294 \\ 4,2996 \cdot 10^{-6} & -4,2001 \cdot 10^{-9} & 0,0077 \\ -0,3289 & -0,0080 & 0,8916 \end{bmatrix} \quad (3.6)$$

Algoritmus, pomocou ktorého sú dosiahnuté výsledky v obrázku č. 16, bol vytvorený v programovacom prostredí Matlab. Bol na to využitý modul *Stereo Vision Toolbox*, ktorý sa v praxi využíva na 3D videnie. Podporuje kalibráciu jednej kamery, dvoch alebo kamery s fisheye efektom. Funkcie, ktoré aktuálna verzia knižnice OpenCV nepodporuje, ako napríklad Sift alebo Surf, sú v knižnici Matlabu stále prítomné. Na obrázku č. 16 vpravo je zobrazené a zvýraznené nájdenie spoločných bodov pomocou funkcie *detectSURFFeatures*. V Matlabe niektoré vzorové algoritmy využívajú rektifikáciu a odhad fundamentálnej matice na to, aby dokázali vytvoriť 3D model

z dvoch snímok. Tomu predchádza kalibrácia kamery, ktorá dodáva do algoritmu ďalšie dáta na vypočítanie rotácie a translácie. To vedie ku ďalším krokom, ktoré určia matice kamier, na základe ktorých môže prebehnúť triangulácia a výsledné zistenie 3D bodov. Nakoľko tieto údaje pri meraní pomocou riadkovej kamery dostupné neboli, bolo potrebné vytvoriť vlastný matematický model, ktorý by pomohol určiť rozmery objektu s čo najväčšou presnosťou. Okrem toho boli vypracované výpočty, ktoré by maticu rotácie a translácie poskytli.

3.6.3 3D model vytvorený pomocou hĺbkovej mapy

Výsledky dosiahnuté odhadovaním parametrov pre hĺbkovú mapu boli nepoužiteľné a nepribližovali sa tomu, čomu by sa skutočne mali. Algoritmus mal problém vypočítať hĺbku, resp. skutočné rozmery, ktoré sa v snímkach vyskytovali.



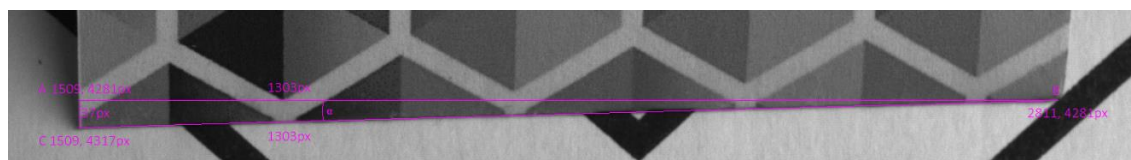
Obr. č. 16: Výsledky 3D merania s použitím hĺbkovej mapy

Rovnaké výsledky, ako na obrázku č. 16, boli dosiahnuté aj vtedy, keď boli body nachádzajúce sa na parabole oddelené od bodov podložky. Cieľom tohto kroku bola lepšia presnosť, pretože by sa nemuseli merať dve plochy naraz, ale iba jedna. Zhoda bielej plochy je na niektorých miestach rovnaká, avšak zhoda hrán je to, čo nesie informáciu, z ktorej sa dá niečo vypočítať. Takúto nepresnosť môže spôsobovať napríklad nepresná detekcia rohov.

3.7 Návrh matematického modelu počítania 3D modelu

3.7.1 Určenie náklonu objektu

V kapitole 3.5 bolo opísané, ako prebieha meranie pomocou riadkovej kamery, aké nepresnosti vznikajú v obraze ako dôležité je presné narábanie s kamerou. Ak nie je kamera nastavená v správnej polohe, pri meraní nemusia byť dosiahnuté správne výsledky a treba na tieto nepresnosti myslieť. Korekcia objektu zo snímky jednej kamery má za úlohu odstrániť nepresnosti v narastajúcej šírke objektu pri zvyšujúcej sa výške. Vďaka zmene šírky je možné merať, o koľko v každom riadku jeho výška stúpla a klesla. Keďže meranie tejto výšky prebieha v rovine pohľadu kamery, je potrebné, aby bol objekt vodorovne. Keby nebol, vznikla by asymetria vo výške objektu v riadku. Tým, že je objekt natočený, zaisťuje, že v každom bode objektu v riadku je výška konštantná.



Obr. č. 16: Vychýlenie meraného objektu od roviny pohľadu

V obrázku č. 16 je riadok predstavujúci rovinu pohľadu – horná ružová priamka – a spodná hrana objektu je to, čo je vychýlené od pohľadu a treba túto výchylku určiť.

Vzdialenosť medzi tromi vyznačenými bodmi je možné vypočítať pomocou vektorov. Vzďialenosť medzi ľavým horným bodom a bodom napravo je vypočítaná nasledovne:

$$|\overline{AB}| = \sqrt{(b_1 - a_1)^2 + (b_2 - a_2)^2} \quad [\text{px}] \quad (3.7)$$

$$|\overline{AB}| = \sqrt{(2811 - 1509)^2 + (4281 - 4281)^2} \quad [\text{px}] \quad (3.8)$$

$$|\overline{AB}| = 1302 \quad [\text{px}] \quad (3.9)$$

Výsledná veľkosť vektoru je 1302 pixelov. V obrázku č. 16 sa nachádza orientačná dĺžka jednotlivých vektorov, v ďalších výpočtoch sú však využité manuálne dopočítané hodnoty. Takýmto spôsobom je možné overiť veľkosť všetkých priamok, ktoré boli v obrázku č. 16 pre výpočet výchylky určené. Dosadením súradníc do výpočtu 3.7 ľavého horného bodu a ľavého spodného bodu určí, o koľko pixelov sa v danom mieste roviny od seba vychýľujú:

$$|\overline{AC}| = \sqrt{(1509 - 1509)^2 + (4317 - 4281)^2} \quad [\text{px}] \quad (3.10)$$

$$|\overline{AC}| = 36 \quad [\text{px}] \quad (3.11)$$

S týmito vypočítanými hodnotami je ďalej možné určiť, o koľko stupňov sa roviny vychýľujú. Uhol α je vypočítaný nasledovne:

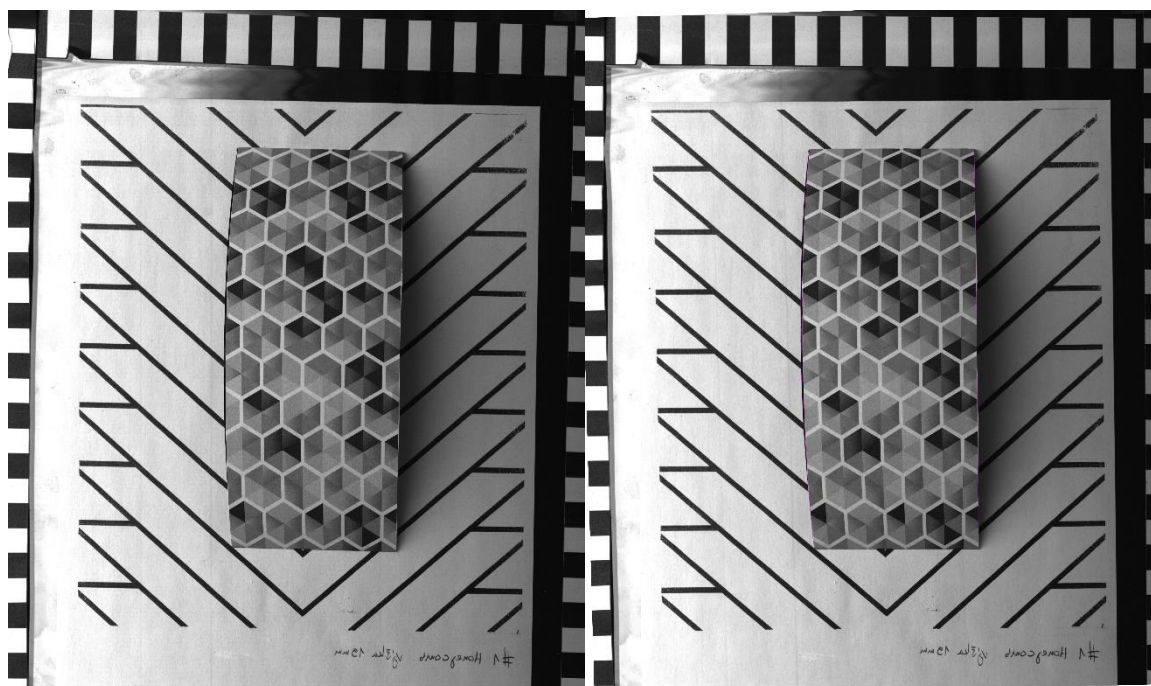
$$\alpha = \tan^{-1} \left(\frac{|\overline{AB}|}{|\overline{AC}|} \right) \quad [\text{rad}] \quad (3.12)$$

$$\alpha = \tan^{-1} \left(\frac{36}{1302} \right) \quad [\text{rad}] \quad (3.13)$$

$$\alpha = 1.5838 \quad [\text{rad}] \quad (3.14)$$

$$\alpha = 1^\circ 35' \quad [^\circ] \quad (3.15)$$

Rovnakým spôsobom sa určuje aj náklon medzi rovinou pohľadu a podložkou, na ktorej objekt bol objekt meraný. Teraz je potrebné tento náklon odstrániť. Existujú algoritmy, ktoré ponúkajú široký výber možností, ako obrázky otočiť. Pri otáčaní obrázkov vznikajú v snímkach biele miesta, na ktorých sa predtým vyskytovali pixely. Posunutím pôvodných pixelov na iné miesto tu teda vzniká možnosť, pomocou ktorej je možné určiť, či majú byť nové pixely konkrétnej farby alebo budú priehľadné. Priehľadné pixely je možné aplikovať iba pri snímkach vo formáte .PNG. Otočením obrázka treba myslieť na jeho výsledné rozlíšenie. V tomto prípade bolo zachované pôvodné rozlíšenie obrázka, aby bolo možné pracovať so skutočnými rozmermi, ktoré boli meraním dosiahnuté. Okrem týchto možností je možné zvoliť, okolo ktorého bodu v obrázku sa má plocha rotovať a či sa má rozlíšenie upraviť tak, aby sa odrezali nové miesta, ktoré by boli prázdne. Práca je vykonávaná so snímkami vo formáte .PNG kvôli menšej strate na kvalite a presnosti merania. Výsledok rotácie snímky v obrázku č. 17, na ktorom vidno, že horná hrana meraného objektu je natočená vodorovne so spodnou hranou meraného objektu. V tomto bode merania nastáva posun ku korekcii objektu, počas ktorej sa budú počítať rozmery meraného objektu pre vyhotovenie 3D modelu.



Obr. č. 17: Výsledok rotácie snímky (vpravo) a pôvodná snímka (vľavo)

3.7.1 Korekcia skreslenia objektu v snímke z jednej kamery

V kapitole 3.5.2 bola opísaná detekciu hrán objektu, ktorá vedela na miesta hrán vkladať ružové pixely a tým označiť, čo je meraný objekt. Na meranej parabole bol bledý vzor so šesťuholníkmi. Tento vzor sa však dá ľahko vymeniť za iný a niektoré z týchto vzorov majú mierne opotrebované hrany. Ak nie je vzor presne umiestnený, môže vzniknúť odchýlka vo výpočte výšky. To je vidieť na obrázku č. 17, kde na ľavej hrane vyčnieva malá časť pôvodného objektu a vzor ju nezakrýva správne. Táto odchýlka kriticky ovplyvňuje meranie, čo vidno aj vo výslednom 3D modeli z nameraných dát z jednej kamery. Algoritmus počíta rozmery objektu v snímke, ktorá má upravený pomer strán a je vo formáte .PNG. Prechádza každý pixel v každom stĺpci. Na začiatku algoritmu korekcie sú zadefinované polia, do ktorých sú v priebehu algoritmu vkladané zmerané dáta. Detekcia začína tak, že každý pixel je porovnávaný s RGB hodnotami (255, 0, 255), čo sú hodnoty ružovej farby. Ak nie je splnená podmienka rovnosti s touto hodnotou, algoritmus pokračuje ďalej v porovnávaní pixelov. Pre lepšiu detekciu bolo odhadnuté, v akom rozmedzí pixelov (počet stĺpcov pre bočné hrany a počet riadkov pre hornú a spodnú hranu) sa ružové pixely nachádzajú. Ak algoritmus prvýkrát natrafi v riadku na ružový pixel, začína počítanie, koľko pixelov sa za ružovým pixelom nachádza. Ak počítanie príde do hranice odhadovaného ďalšieho kraja objektu a narazí na ďalší ružový pixel, šírka objektu v pixeloch je tým úspešne zmeraná. Číslo stĺpca, v ktorom sa druhý nájdený ružový pixel nachádza, sa odráta od celkovej šírky snímky v pixeloch. Tento výpočet určí, koľko pixelov sa nachádza v riadku napravo od objektu a následne dopočíta, koľko pixelov sa nachádza aj naľavo od objektu. Táto časť algoritmu pracuje s obrázkom v rozlíšení 4600px v osi Y a 5434px v osi X a zaberie niekoľko minút. Ďalej nasleduje spracovanie zmeraných údajov.

Pri meraní pomocou algoritmov na spracovanie obrázkov bolo zistené, že spodná hrana má šírku 1302 pixelov. Rovnakým spôsobom je možné určiť aj šírku vrchnej hrany, pretože merací algoritmus začína indexovať od súradníc $X=0$ a $Y=0$ a tie sa nachádzajú v ľavom hornom rohu obrázka. Vrchná hrana objektu je teda prvá, na ktorej začne algoritmus počítať rozmery. Je potrebné vypočítať, v akom pomere je táto hodnota s ďalšími hodnotami z ďalších riadkov. Vykoná sa séria výpočtov, ktoré zistia na základe

vypočítaného pomeru, ako zmeniť šírku obrázka v danom riadku tak, aby boli rozmery objektu opravené.

$$pomer = \frac{\text{šírka objektu}_{\text{prvý riadok}}}{\text{šírka objektu}_{\text{meraný riadok}}} \quad [\text{px}] \quad (3.16)$$

$$\text{nová šírka riadka} = \frac{\text{šírka obrázka}}{pomer} \quad [\text{px}] \quad (3.17)$$

Zmeraná šírka vrchnej hrany v zrotovanom obrázku je 1310 pixelov. V príklade výpočtu sú uvedené zmerané hodnoty z riadku 3063, v ktorom je objekt široký 1381 pixelov. Dosadením týchto hodnôt do rovnice 3.16 je možné vypočítať pomer, v akom bude potrebné zmeniť šírku snímku pre daný riadok na vykonanie korekcie.

$$pomer = \frac{1381}{1310} \quad [-] \quad (3.18)$$

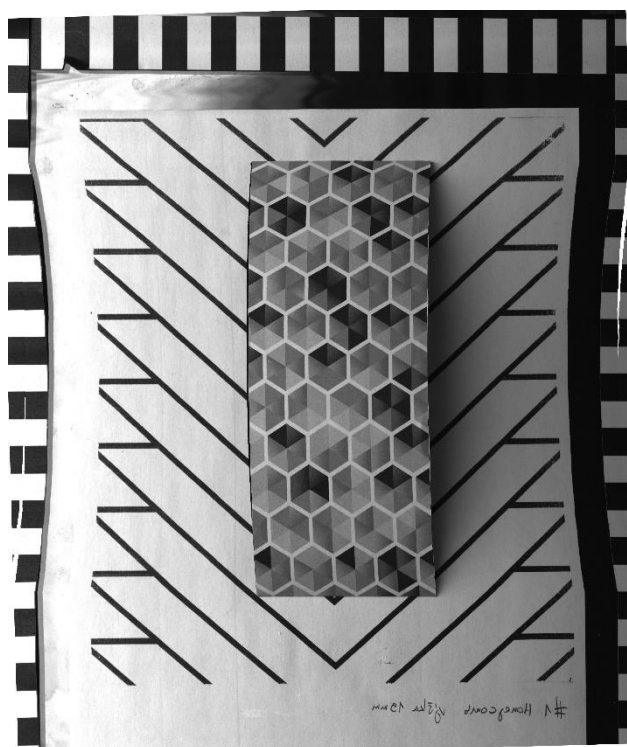
$$pomer = 1,0542 \quad [-] \quad (3.19)$$

Tento pomer udáva, ako veľmi je zväčšený objekt oproti svojej pôvodnej veľkosti. V rovnakom pomere treba zmenšiť šírku riadka, aby prebehlo odstránenie skreslenia. Zmena šírky, teda rozlíšenia, vykonáva funkcia *resize*.

$$\text{nová šírka obrázka} = \frac{4600}{1,0542} \quad [\text{px}] \quad (3.20)$$

$$\text{nová šírka obrázka} = 4353,5047 \quad [\text{px}] \quad (3.21)$$

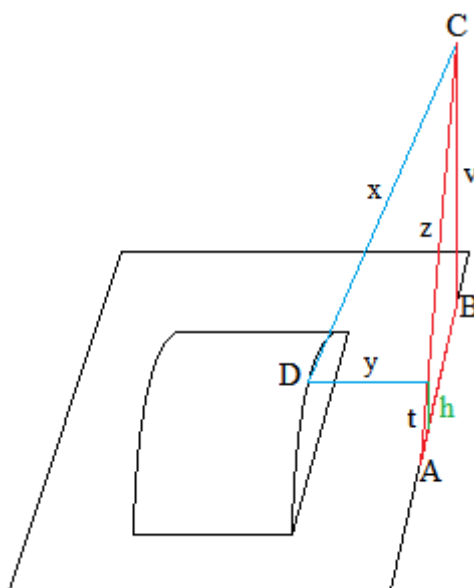
$$\text{nová šírka obrázka} \doteq 4354 \quad [\text{px}] \quad (3.23)$$



Obr. č. 17: Výsledok kalibrácie objektu v snímke

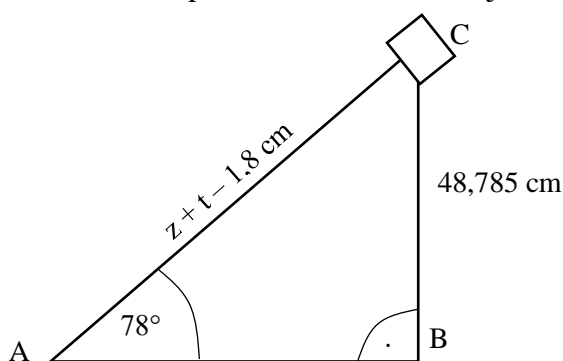
3.7.2 Počítanie výšky objektu

Dáta zmeraná v prvých častiach algoritmu sa uložia a ďalej sa pokračuje v ich spracovaní. Dáta sú spracúvané podľa nasledujúceho matematického modelu:



Obr. č. 18: Matematický model výpočtu rozmerov objektu

Výška kamery je 45,875 cm a je označená písmenom v . Uhol, pod ktorým bol snímaný objekt je 78° . To umožňuje dopočítať stranu, na ktorej sa nachádzajú dĺžky t a z . Od tejto strany je potrebné odrátať rozmer kamery. Bod C predstavuje stred kamery, to znamená, že ak je šírka kamery 36 milimetrov, polovica z tohto zasahuje do dĺžky pohľadu.



Obr. č. 19: Počítanie dĺžky pohľadu

Na určenie dĺžky rovina pohľadu slúži sínusova vetu. Do nej sú vložené všetky známe údaje z obrázka č. 19. Polovičný rozmer kamery zatiaľ odrátať netreba, tento úkon je možné vykonať na záver počítania.

$$dĺžka\ pohľadu = v \cdot \frac{\sin(\beta)}{\sin(\alpha)} \quad [cm] \quad (3.24)$$

$$dĺžka\ pohľadu = 48,785 \cdot \frac{\sin(90^\circ)}{\sin(78^\circ)} \quad [cm] \quad (3.25)$$

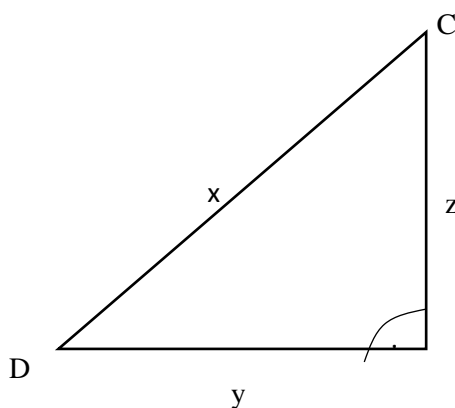
$$dĺžka\ pohľadu = 46,9224 \quad [cm] \quad (3.26)$$

Rovnica 3.24 meria dĺžku pohľadu. V ďalšom kroku je potrebné vypočítať ohniskovú vzdialenosť v pixeloch, ktorá je potrebná na určenie vzdialenosti medzi kamerou a objektom v každom riadku, v ktorom sa nachádza.



Obr. č. 20: Šachovnica nad objektom

Na obrázku č. 20 sa nachádza šachovnica, ktorá určuje, že 7 centimetrov sa rovná 1270 pixelom. Ak je toto číslo vydelené počtom 7 centimetrov, určí sa, že jeden centimeter má približne 181,4286 pixelov. Tento údaj je zadefinovaný preto, lebo meraný objekt má šírku 7 centimetrov a číslo v pixeloch sa ďalej využíva pri počítaní. Počas merania sa viacero parametrov menilo a to viedlo k manuálnemu overovaniu výsledkov. Počas merania prebehlo niekoľko pokusov zaostriť kameru, čo malo za následok zmenu ohniskovej vzdialenosti. Kamera sa nachádzala v rozmedzí od 40 do 50 cm. Najpresnejšie výsledky merania vychádzajú, ak je výška kamery 45,897 cm a ohnisková vzdialenosť 8588,194 pixela.



Obr. č. 21: Počítanie vzdialenosti medzi prekážkou a kamerou

$$x = \frac{W \cdot F}{P} \quad [\text{cm}] \quad (3.27)$$

$$x = \frac{7 \cdot 8588,194}{1387} \quad [\text{cm}] \quad (3.28)$$

$$x = 43,3434 \quad [\text{cm}] \quad (3.29)$$

Výpočtom rovnice 3.28 je získaná vzdialenosť medzi objektom a kamerou, teda 43,3434 centimetra. Číslo 1387 je šírka objektu v riadku, v ktorom by mal dosahovať svoje maximum, ktoré je 2,5 centimetra. Tento údaj bol využitý pri kalibrácii sústavy na odhadnutie ohniskovej vzdialenosti a výšky, v ktorej sa kamera nachádza. Nebol to však údaj, od ktorého by sa postupne výpočty odvodzovali. Číslo 7 predstavuje skutočnú šírku objektu v centimetroch. Rozmer t v obrázku č. 18 je doložený tak, že od dĺžky pohľadu

je odrátaný polovičný rozmer kamery (1,8 cm) a parameter z . Parameter z je opäť možné určiť pomocou kosínusovej vety nasledovným spôsobom:

$$x^2 = y^2 + z^2 - 2yz \cos(\beta) \quad [\text{cm}] \quad (3.30)$$

$$x^2 = y^2 + z^2 - 2yz \cos(90^\circ) \quad [\text{cm}] \quad (3.31)$$

$$x^2 = y^2 + z^2 - 2yz \cdot 0 \quad [\text{cm}] \quad (3.32)$$

$$z^2 = x^2 - y^2 \quad [\text{cm}] \quad (3.33)$$

$$z = \sqrt{x^2 - y^2} \quad [\text{cm}] \quad (3.34)$$

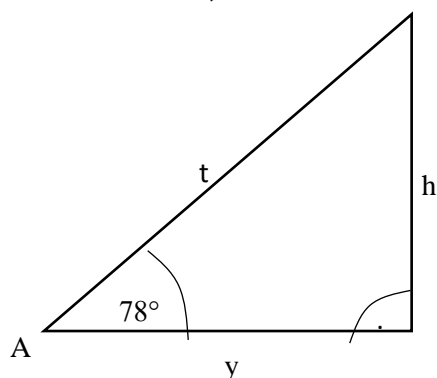
$$z = \sqrt{43,3434^2 - 8,1795^2} \quad [\text{cm}] \quad (3.35)$$

$$z = 42,5647 \quad [\text{cm}] \quad (3.36)$$

Teraz dopočítame dĺžku t :

$$t = 46,9224 - 1,8 - 42,5647 \quad [\text{cm}] \quad (3.37)$$

$$t = 2,5577 \quad [\text{cm}] \quad (3.38)$$



Obr. č. 22: Počítanie výšky objektu

Na základe obrázka č. 22 je zrejmé, že sú k dispozícii všetky potrebné údaje na dopočítanie výšky h . Na jej výpočet aplikujeme opäť sínusovú vetu podobne, ako v kroku (3.24)

$$h = t \cdot \frac{\sin(78^\circ)}{\sin(90^\circ)} \quad [\text{cm}] \quad (3.39)$$

$$h = 2,5577 \cdot \frac{\sin(78^\circ)}{\sin(90^\circ)} \quad [\text{cm}] \quad (3.40)$$

$$h = 2,5018 \quad [\text{cm}] \quad (3.41)$$

Vzťah 3.41 ukazuje, že vypočítaná výška sa líši od reálnej iba o 0,0018 centimetra. Algoritmus, ktorý meria výšku objektu na základe šírky, bol najprv testovaný na šachovnici z obrázka č. 20. Na základe údajov z tohto merania sa zistilo, koľko pixelov predstavuje 7 centimetrov a pre túto hodnotu bola odhadnutá ohnisková vzdialenosť, ktorá ďalej slúžila na meranie výšky.

3.8 Tvorba 3D modelu

3.8.1 Postup pri tvorbe algoritmu vykresľovania

V predchádzajúcich kapitolách bol popísaný algoritmus, ktorý upravuje rozmery skresleného objektu. Počas počítania rozmerov boli zistené potrebné informácie pre vykreslenie scény z jednej kamery. Kapitola 3.5 sa zaoberala meraním pomocou reálnej riadkovej kamery a bližšie popísala nepresnosti, ktoré sa môžu v sústave vyskytnúť. Všetky tieto nepresnosti vplývajú aj na výsledný obraz. Časť týchto nepresností odstraňuje vytvorený korekčný algoritmus, niektoré ďalšie je však potrebné manuálne dopočítať. Jedna sa výpočet matice rotácie alebo translácie, ktoré popisujú, k akým zmenám v snímku dochádza a ako sa jednotlivé body posúvajú. Táto kapitola popisuje algoritmus, ktorý vykresľuje všetky zmerané dáta do 3D modelu z jednej kamery a priblíži jednoduchú geometriu, akou sa merajú posuny bodov medzi dvoma snímkami. 3D model z jednej kamery má za úlohu predviesť, aké nepresnosti sa v sústave nachádzajú. Všetky tieto nepresnosti sa dajú matematicky vyriešiť, výpočty sú však komplikovanejšie a je potrebné, aby bol k dispozícii väčší počet snímkov pre spresnenie merania.

V rovnici 3.41 sa nachádza výška, ktorá bola dopočítaná pre daný riadok snímku. Výška objektu vyšla 2,5018 centimetra a odchýlka od reálnej hodnoty je iba 0,0018 centimetra. Pri tvorbe 3D modelu je potrebné toto číslo prepočítať na pixely, pretože výsledný 3D model pracuje s pôvodnými súradnicami X a Y v pixeloch a výška Z by bola oproti týmto súradniciam veľmi malá. Na začiatku algoritmus je definovaných niekoľko polí, do ktorých sú v ďalších krokoch ukladané nové dáta potrebné pre vykreslenie 3D modelu.. Sú to polia pre súradnicu X, pre súradnicu Y, výšku Z a farebné zložky R, G a B. Na zistenie farby pixelu je potrebné PIL. Je to formát, ktorý ponúka knižnica OpenCV a s obrázkami v tomto formáte sa pracuje inak. Obsahuje iné funkcie, ako modul `cv2` a preto sa mení aj syntax písania kódu.

Pre uľahčenie výpočtového výkonu sú všetky doposiaľ namerané údaje zmenšené o tretinu. To urýchli ďalšie počítanie, pretože teraz sa nové údaje nemerajú, ale spracúvajú už tie namerané. Vykresľovanie 3D modelu nebude graficky náročné, pretože nebude musieť zobrazovať príliš veľa informácií. Po načítaní snímky vo formáte PIL sú zadefinované premenné, v ktorých je šírka a výška snímky v pixeloch zmenšená o tretinu. Funkciou `resize` je následne rozlíšenie zmenené. Takto upravená snímka je potom prekonvertovaná na RGB farebný model. Aj napriek tomu, že pôvodná snímka bola čiernobiela, tieto farby sa z jedného kanálu rozdelia na tri kanály, teda tri farebné zložky. Vďaka tomu je možné pomocou funkcie `getpixel` zistiť, akú hodnotu má každý kanál, resp. farebná zložka.

Polia, do ktorých sa údaje o snímke ukladajú, majú rozmer *šírka·výška* upravenej snímky. Je to preto, lebo tieto polia sú jednorozmerné a je potrebné, aby napríklad súradnica X, bola uvedená pre každý pixel v snímke. Znamená to, že ak bola pôvodná snímka široká 4600 pixelov, tretina tejto výšky je zaokrúhlene 1533 pixelov. Počítadlo, ktoré naplňa pole súradnicami X, musí napočítať od 0 do 1532 až 1811 krát a každé číslo zapisovať. Pri dosiahnutí čísla 1532 sa počítanie resetuje a začne od začiatku. Rovnaký princíp je aplikovaný aj pri naplňaní poľa so súradnicou Y.

Pri plnení poľa so súradnicou Z, resp. s výškou objektu, je potrebné počítať inak. Na základe znalostí, v ktorých riadkoch sa meraný objekt nachádza a koľko pixelov zaberajú strany po bokoch od objektu, je možné určiť, že pre túto oblasť mimo objektu je výška rovná 0, pretože to je podložka a od tej vzrastá objekt do výšky. Keďže boli rozmery snímky zmenšené o tretinu a zmerané dáta poskytujú údaje pre 3150 riadkov, treba

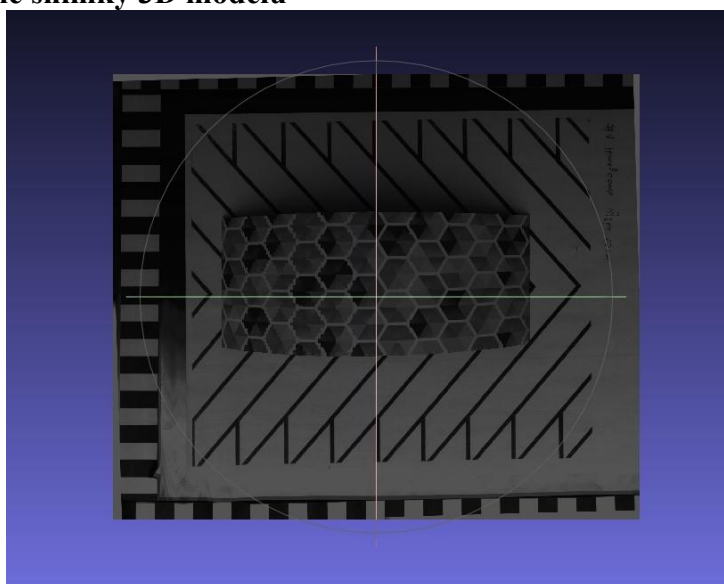
vybrať každú tretiu hodnotu a vložiť ju do samostatného poľa. To isté je spravené aj s údajmi o vzdialenostiach naľavo a napravo od objektu. Tieto nové polia budú mať oproti svojej pôvodnej veľkosti dĺžku 1046 riadkov. Jedno počítaadlo počíta od 0 do 1046 a ďalšie počíta od 0 a k tejto hodnote pripočítava číslo 3.

Pri plnení poľa so súradnicou Z sa určí, v ktorých častiach snímky sa objekt nachádza. To určia údaje o vzdialenosti medzi objektom a krajom obrázka po jeho stranách tak, že sa na ich základe určí súradnica Y, ktorá tvorí hranu objektu. Ďalší cyklus, prechádza po každom pixeli v snímke a na základe zmeraných údajov sa volí, ktorý pixel má výšku 0 a ktorý má výšku zmeranú. Na tvorbu 3D modelu je možné použiť snímky s opraveným objektom, no na demonštráciu scény bol zvolený zrotovaný obrázok, ktorý poskytne predstavu o nepresnosti sústavy z kapitoly 3.5.1.

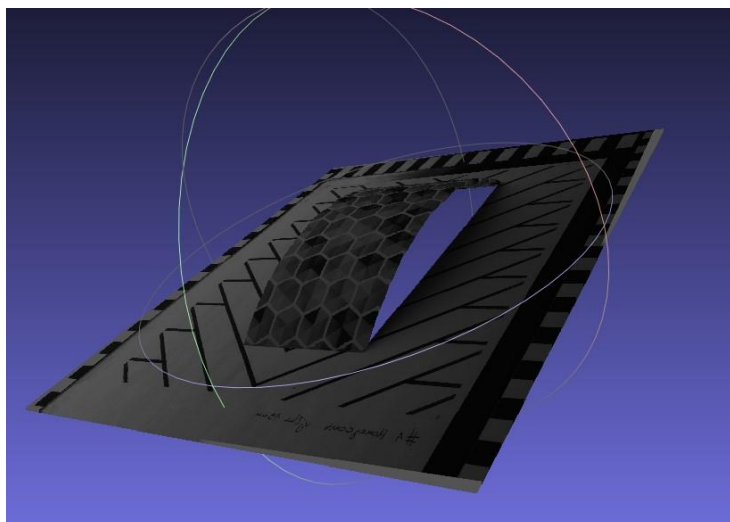
Existuje niekoľko spôsobov, ako výsledné zmerané dáta uložiť a ako ich zobrazit'. Na zobrazenie 3D modelu bol použitý program MeshLab, ktorý dokáže spracovať široké množstvo 3D modelov v rôznych formátoch. Nezobrazuje nielen jednotlivé body v X, Y a Z súradniciach, zobrazuje aj farebné zložky pixelu.

Teraz, keď sú dostupné všetky potrebné údaje, je možné ich uložiť vo formáte .txt. Výsledný 3D model z jednej kamery je vo formáte X Y Z R G B, k dispozícii sú však aj mnohé ďalšie, ktoré program MeshLab dokáže spracovať. Váhu má aj to, ako sú jednotlivé súradnice a zložky vo výstupnom súbore vykresľovacieho algoritmu oddelené. V tomto prípade sú oddelené medzerou a polia so spracovanými údajmi označujú svoje údaje ako *vertices*, v preklade *vrcholy*. Tieto polia majú dĺžku 2 777 378 riadkov.

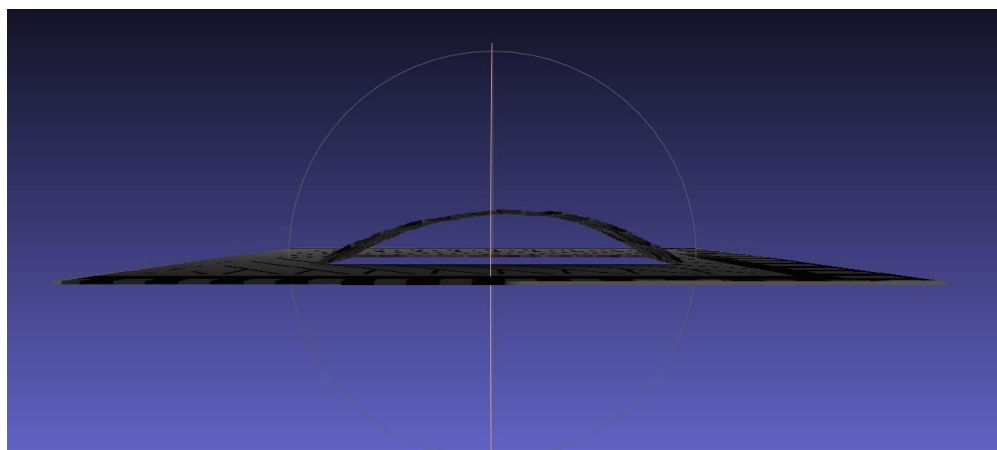
3.8.2 Výsledné snímky 3D modelu



Obr. č. 23: Pohľad na model zhora

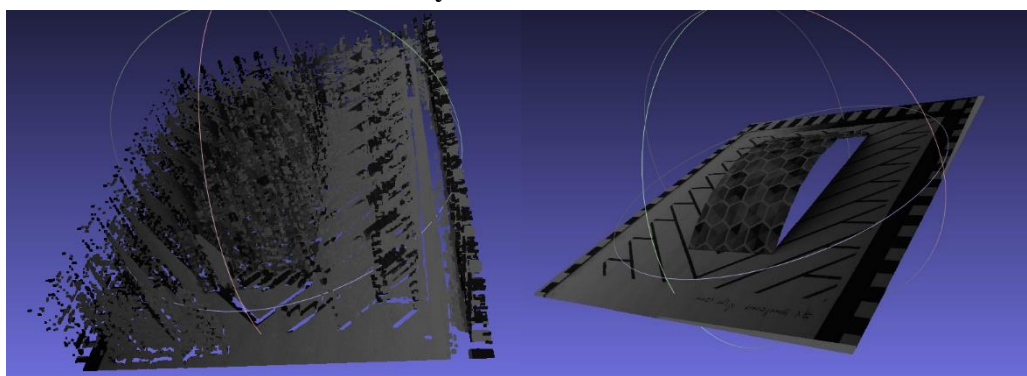


Obr. č. 24: Pohľad na model z boku



Obr. č. 23: Pohľad na model zprava

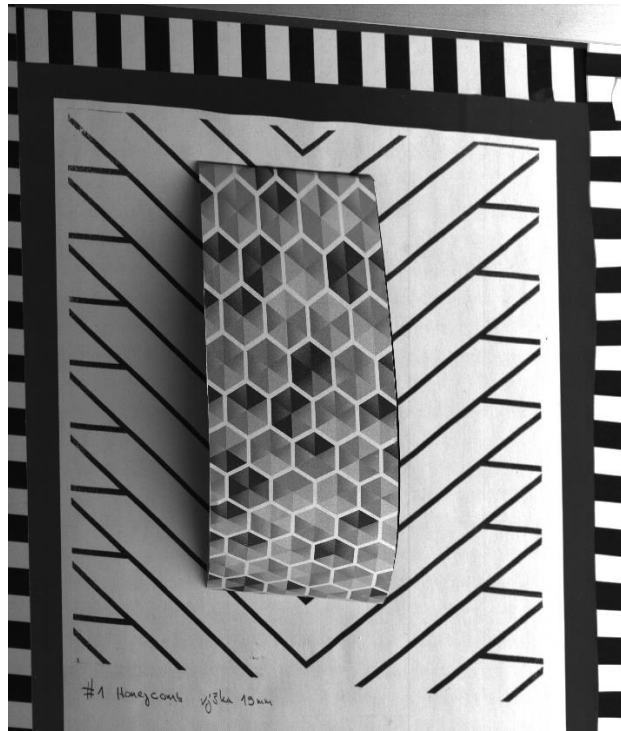
3.8.3 Zhodnotenie 3D modelu scény



Obr. č. 24: Porovnanie 3D modelov

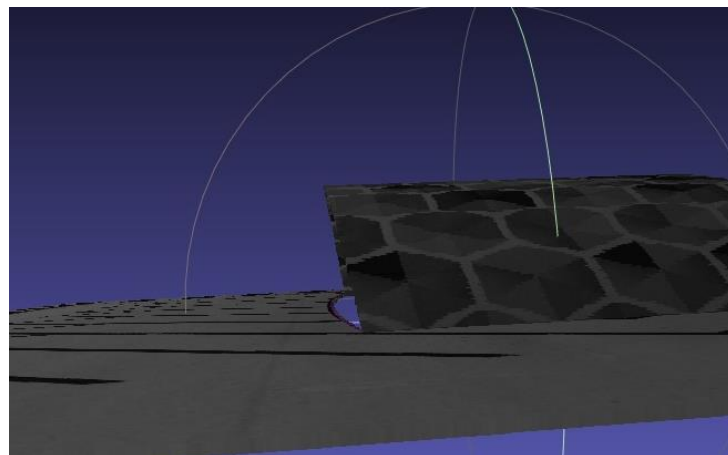
Na obrázku č. 24 je možné vidieť porovnanie vypočítaného modelu (vpravo) a modelu vzniknutého na základe hĺbkovej mapy (vľavo), ktorá stráca takmer všetky informácie o reálnych rozmeroch. Na obrázku č. 23 sú zobrazené dve nepresnosti. Tá prvá je, pravá časť snímky je naklonená a je nižšie, ako ľavá časť. Spôsobuje to náklon kamery, pod ktorým bol objekt snímaný. Snímka 3D modelu bola vyhotovená pod uhlom 78° . Ak by však tento uhol bol menší, resp. viacnásobne väčší, ako napríklad na snímku č. 25, ktorý predstavuje uhol 135° , táto asymetria by bola väčšia. Tieto nepresnosti je

možné odstrániť vďaka znalosti, pod akým uhlom bola snímka vyhotovená. Vyžaduje to však zložitejšie vzorce podobné tým, ktoré určili náklon roviny.



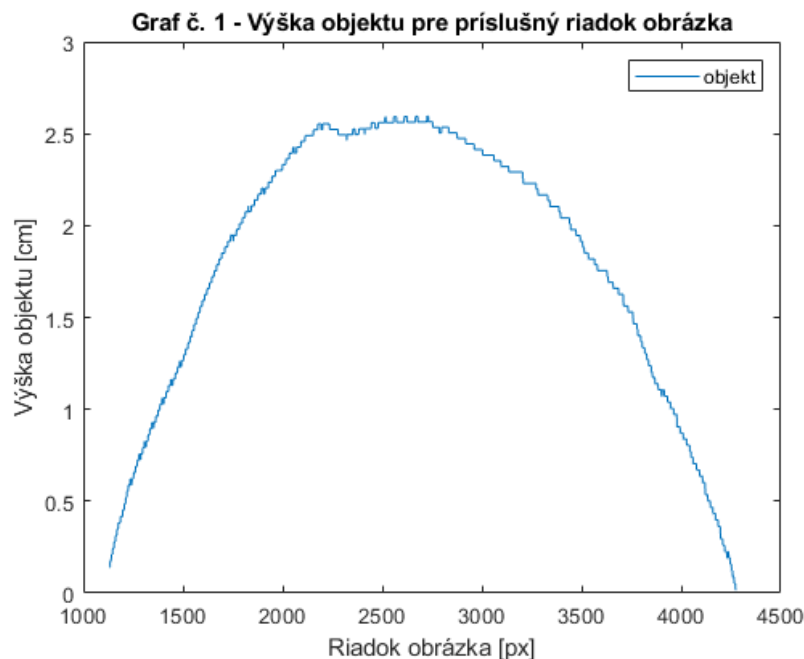
Obr. č. 25: *Asymetria výšok objektu vzniknutá náklonom kamery*

Na obrázku č. 25 rovnako vidieť aj náklon roviny a objektu, ktoré treba odstrániť rovnakým spôsobom, ako v kapitole 3.7.1. Tieto náklony sú dôležité pri zisťovaní posunu jednotlivých pixelov medzi dvoma snímkami, pretože pri každej manipulácii s riadkovou kamerou sú dosiahnuté rozdielne výsledky, ktoré vyžadujú korekciu. Pri snímaní objektu zo strany vzniká ďalšia chyba, ktorá je na obrázku č. 26.

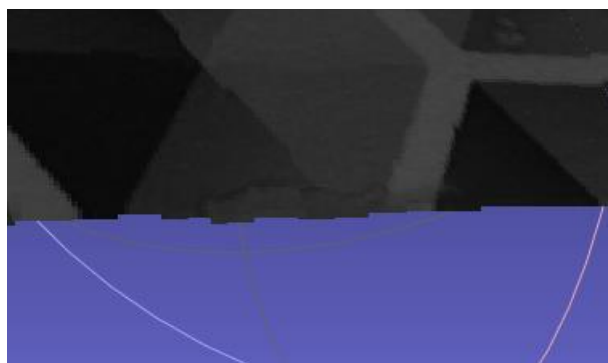


Obr. č. 26: *Nepresnosť vzniknutá snímaním zo strany*

Na obrázku č. 26 je vidieť, že kamera nebola umiestnená kolmo nad meraným objektom, ale bola vychýlená do strany. To spôsobuje, že vzdialenosť medzi pixelmi nie je rovnaká. S rastúcou vzdialenosťou od kamery sa tvary predlžujú, čo spôsobilo nepresnosť znázornenú v spomínanom obrázku. Toto by bolo možné odstrániť na základe znalostí homografie a ďalšie parametrov kamery, ktoré by boli výstupom kalibračných funkcií kamery.



Graf č. 1 ukazuje nepresnosť pri meraní výšky. Medzi riadkami 2000 až 2500 nastala niekoľkomilimetrová výchylka, ktorú spôsobili mierne poškodené hrany pripevneného vzoru na meranom objekte (obrázok č. 27). Táto výchylka sa preukázala aj vo výslednom 3D modeli na obrázku č. 23. Okrem tejto odchýlky tu vidno aj asymetriu výšok, ktorá bola popísaná v predchádzajúcich častiach práce a súvisela aj s obrázkom č. 25.

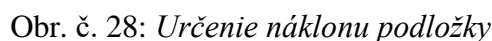


Obr. č. 27: Poškodenie vzoru ovplyvňujúce zmeranie výšky objektu

3.9 Posuny dvoch snímok

3.9.1 Odstránenie výchylky rovín prvej snímky

V predchádzajúcich kapitolách bolo rozobrané, aké informácie nesie jedna snímka z jednej kamery. Matematický model, ktorý vypočítal výšku objektu v každom riadok, v ktorom sa meraný objekt nachádzal, bral údaje zo snímky vyrotovanej o uhol $1^{\circ}54'$. Bolo to potrebné preto, aby sa odstránila asymetria výšok v objekte pre meraný riadok. Takáto odchýlka môže vzniknúť v meracej sústave a je potrebné ju odstrániť, resp. s ňou pracovať vo výpočtoch. Postup je podobný ako v kapitole 3.7.1.



Body A				Body B			
bod	súradnice	[px]	[mm]	bod	súradnice	[px]	[mm]
A1	x	4270	235,35	B1	x	4175	230,12
	y	499	27,5		y	4039	222,62
A2	x	4614	254,32	B2	x	4554	251,01
	y	507	27,45		y	4045	222,95
A	x	4522		B	x	4522	
	y	505			y	4044	

$$||A_1 A_2|| = \sqrt{(A_{1X} - A_{2X})^2 + (A_{1Y} - A_{2Y})^2} \quad [\text{px}] \quad (3.42)$$

$$||A_1 A_2|| = 344,093 \quad [\text{px}] \quad (3.43)$$

$$||A_1 A|| = \sqrt{(A_{1X} - A_X)^2 + (A_{1Y} - A_Y)^2} \quad [\text{px}] \quad (3.44)$$

$$||A_1 A|| = 252,071 \quad [\text{px}] \quad (3.45)$$

$$t_A = \frac{||A_1 A||}{||A_1 A_2||} = 0,733 \quad [\text{px}] \quad (3.46)$$

$$\tan \alpha = \frac{|A_X - B_X|}{|A_Y - B_Y|} \quad [\text{rad}] \quad (3.47)$$

$$\alpha = \frac{|A_X - B_X|}{|A_Y - B_Y|} = 0,934 \quad [\text{rad}] \quad (3.48)$$

Uhol α udáva výchylku, ktorá je znázornená v obrázku č. 28 medzi úsečkami $|AC|$ a $|AA_2|$. Tento uhol je však možné nájsť aj medzi úsečkami $|A_1B_1|$ a $|AB|$. V tomto kroku nasleduje výpočet súradníc bodu C. Pre porovnanie a lepšie chápanie výsledných výpočtov sú uvedené výsledky nielen v jednotkách pixelov, ale aj v milimetroch:

$$C_X = A_X + d_X \quad [\text{px}] \quad (3.49)$$

$$C_X = 4522 + 61 \quad [\text{px}] \quad (3.50)$$

$$C_X = 4583 \quad [\text{px}] \quad (3.51)$$

$$C_X = 252,606 \quad [\text{mm}] \quad (3.52)$$

Výsledok 3.51 udáva výpočet súradnice X, ktorá je vo vychýlenej rovine rovnobežnej so vzorom podložky. Je to prvá súradnica posunu, ktorý by sa musel pre bod A vykonať. Nasleduje výpočet súradnice Y pre bod C:

$$C_Y = A_Y + d_Y \quad [\text{px}] \quad (3.53)$$

$$d_Y = \tan \alpha \cdot d_X = 56,956 \quad [\text{px}] \quad (3.54)$$

$$C_Y = 505 + 56,956 \quad [\text{px}] \quad (3.55)$$

$$C_Y = 561,9556 \quad [\text{px}] \quad (3.56)$$

$$C_Y = 30,974 \quad [\text{mm}] \quad (3.57)$$

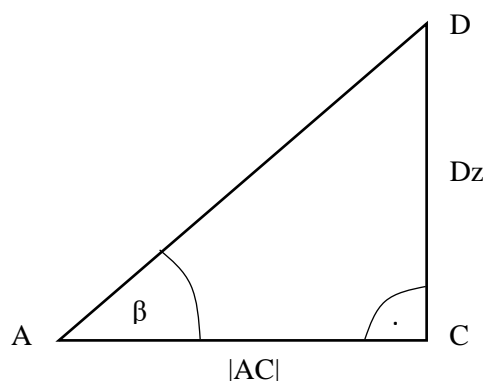
Vzdialenosť medzi bodom A a C je možné vypočítať nasledovným spôsobom:

$$||AC|| = \sqrt{(C_X - A_X)^2 + (C_Y - A_Y)^2} \quad [\text{px}] \quad (3.58)$$

$$||AC|| = 83,487 \quad [\text{px}] \quad (3.59)$$

$$||AC|| = 4,6012 \quad [\text{mm}] \quad (3.60)$$

Výchylka kamery môže niekedy spôsobiť aj nepresnosť v Z-ovej súradnici bodu. Ak by bol náklon kamery do strany až príliš veľký, táto chyba v spojení s výchylkou podobnej tej, ktorá je vypočítaná vo výsledku 3.48, by spôsobila, že by bolo potrebné vykonať aj korekciu výšky daného bodu.



Obr. č. 29: *Finálna korekcia posunu bodu*

Na obrázku č. 29 sa vyskytuje uhol snímania β , ktorý je pri danej snímke veľký 78° . Výšku D_z je možné vypočítať nasledovným spôsobom:

$$D_z = ||AC|| \cdot \tan \beta \quad [\text{px}] \quad (3.61)$$

$$D_z = 405,855 \quad [\text{px}] \quad (3.62)$$

$$D_z = 22,37 \quad [\text{mm}] \quad (3.63)$$

Kompletné výsledky bodu D sú:

$$D_X = C_X = 252,606 \quad [\text{mm}] \quad (3.64)$$

$$D_Y = C_Y = 27,863 \quad [\text{mm}] \quad (3.65)$$

$$D_z = 22,37 \quad [\text{mm}] \quad (3.66)$$

3.9.2 Určenie rotácie a translácie prvej snímky

Rovnaký výpočet na korekciu posunutia bodu je potrebné vykonať aj pre bod na druhej snímke kamery. So získanými súradnicami pre body A, B a D, nachádzajúce sa v jednej rovine, je možné pre každú kameru určiť rotačnú maticu a vektor translácie.

$$\begin{bmatrix} A_X & A_Y & A_Z \\ B_X & B_Y & B_Z \\ D_X & D_Y & D_Z \end{bmatrix} \begin{bmatrix} a_p \\ b_p \\ c_p \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix} \quad [\text{mm}] \quad (3.67)$$

$$M \begin{bmatrix} a_p \\ b_p \\ c_p \end{bmatrix} = D_1 \quad [\text{mm}] \quad (3.68)$$

$$\begin{bmatrix} a_p \\ b_p \\ c_p \end{bmatrix} = M^{-1} D_1 \quad [\text{mm}] \quad (3.69)$$

Rotačná matica M po dosadení vypočítaných čísel a vektor translácie t vyzerajú nasledovne:

$$M = \begin{bmatrix} 249,244 & 27,835 & 0 \\ 250,898 & 222,898 & 0 \\ 252,606 & 27,863 & 22,37 \end{bmatrix} \quad [\text{mm}] \quad (3.70)$$

$$t = \begin{bmatrix} 0,0040 \\ -0,0000 \\ -0,0006 \end{bmatrix} \quad [\text{mm}] \quad (3.71)$$

Po týchto krokoch je možné určiť rovnicu roviny, ktorá je vychýlená od roviny pohľadu.

$$a_p x + b_p y + c_p z + d = 0 \quad [-] \quad (3.72)$$

Pričom d sa v tomto prípade rovná 1. Po dosadení vypočítaných čísel vyzerá rovnica roviny takto:

$$0,0040x - 0,0000y - 0,0006z + 1 = 0 \quad [-] \quad (3.73)$$

Normalizáciou roviny je možné určiť uhly normály so súradnicovými osami X, Y a Z a vypočítať parameter d .

$$\cos \alpha = \frac{0,0040 \cdot 1 - 0 \cdot 0 - 0,0006 \cdot 0}{\sqrt{0,0040^2 + (-0,0000)^2 + (-0,0006)^2}} = 8^\circ 31' \quad [^\circ] \quad (3.74)$$

$$\cos \beta = \frac{0,0040 \cdot 0 - 0 \cdot 1 - 0,0006 \cdot 0}{\sqrt{0,0040^2 + (-0,0000)^2 + (-0,0006)^2}} = 90^\circ \quad [^\circ] \quad (3.75)$$

$$\cos \gamma = \frac{0,0040 \cdot 0 - 0 \cdot 0 - 0,0006 \cdot 1}{\sqrt{0,0040^2 + (-0,0000)^2 + (-0,0006)^2}} = 98^\circ 31' \quad [^\circ] \quad (3.76)$$

$$d = \frac{1}{\sqrt{0,0040^2 + (-0,0000)^2 + (-0,0006)^2}} = 247,234 \quad [\text{mm}] \quad (3.77)$$

3.9.3 Odstránenie výchylky rovín druhej snímky

Výpočty je potrebné vykonať aj pre druhú snímku, s ktorou sa porovnávajú posuny medzi jednotlivými bodmi.

Tab. č. 2 - Body pre snímku s náklonom 105°							
Body A				Body B			
bod	súradnice	[px]	[mm]	bod	súradnice	[px]	[mm]
A1	x	4236	233,48	B1	x	4436	244,5
	y	506	27,89		y	4026	221,91
A2	x	4581	252,5	B2	x	4816	265,45
	y	513	28,28		y	4029	222,07
A	x	4481		B	x	4481	
	y	511			y	4026	

$$|A_1A_2| = 345,071 \quad [\text{px}] \quad (3.78)$$

$$|A_1A| = 245,071 \quad [\text{px}] \quad (3.79)$$

$$t_{A=} \frac{|A_1A|}{|A_1A_2|} = 0,733 \quad [\text{px}] \quad (3.80)$$

$$C_X = 260,047 \quad [\text{mm}] \quad (3.81)$$

$$C_Y = 272,279 \quad [\text{mm}] \quad (3.82)$$

$$|BC| = 52,031 \quad [\text{mm}] \quad (3.83)$$

$$D_z = -193,826 \quad [\text{mm}] \quad (3.84)$$

Kompletné výsledky bodu D z druhej snímky sú:

$$D_X = 260,047 \quad [\text{mm}] \quad (3.85)$$

$$D_Y = 272,279 \quad [\text{mm}] \quad (3.86)$$

$$D_z = -193,826 \quad [\text{mm}] \quad (3.87)$$

3.9.4 Určenie rotácie a translácie prvej snímky

Rotačná matica M po dosadení vypočítaných čísel a vektor translácie t vyzerajú nasledovne:

$$M = \begin{bmatrix} 246,984 & 28,165 & 0 \\ 249,740 & 221,906 & 0 \\ 260,047 & 272,28 & -193,826 \end{bmatrix} \quad [\text{mm}] \quad (3.88)$$

$$t = \begin{bmatrix} 0,0041 \\ -0,0001 \\ -0,0002 \end{bmatrix} \quad [\text{mm}] \quad (3.89)$$

Rovnica roviny pre druhú snímku je podľa vzťahu 3.72 vyjadrená nasledovne:

$$0,0041x - 0,0001y + 0,0002z + 1 = 0 \quad [-] \quad (3.90)$$

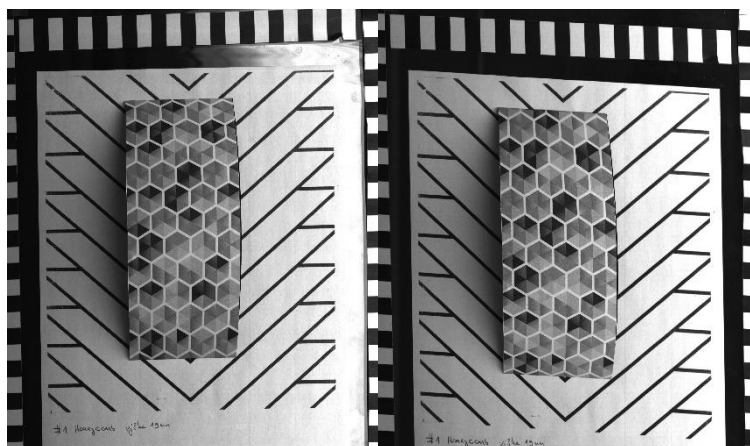
Normalizáciou roviny je možné určiť uhly normály so súradnicovými osami X, Y a Z a vypočítať parameter d .

$$\cos \alpha = \frac{0,0041 \cdot 1 - 0,0001 \cdot 0 + 0,0002 \cdot 0}{\sqrt{0,0041^2 + (-0,0001)^2 + (0,0002)^2}} = 3^\circ 7' \quad [^\circ] \quad (3.91)$$

$$\cos \beta = \frac{0,0041 \cdot 0 - 0,0001 \cdot 1 + 0,0002 \cdot 0}{\sqrt{0,0041^2 + (-0,0001)^2 + (0,0002)^2}} = 91^\circ 23' \quad [^\circ] \quad (3.92)$$

$$\cos \gamma = \frac{0,0041 \cdot 0 - 0,0001 \cdot 0 + 0,0002 \cdot 1}{\sqrt{0,0041^2 + (-0,0001)^2 + (0,0002)^2}} = 87^\circ 12' \quad [^\circ] \quad (3.93)$$

$$d = \frac{1}{\sqrt{0,0040^2 + (-0,0000)^2 + (-0,0006)^2}} = 243,541 \quad [\text{mm}] \quad (3.94)$$



Obr. č. 30: Porovnanie náklonu snímok uhle snímania (78° vľavo, 105° vpravo)

3.9.5 Výsledné korekcie rovín dvoch snímok

$$0,0040x - 0,0000y - 0,0006z + 247,234 = 0 \quad [-] \quad (3.95)$$

$$0,0041x - 0,0001y + 0,0002z + 243,541 = 0 \quad [-] \quad (3.96)$$

V tomto momente sú vypočítané roviny, pomocou ktorých je možné vytvárať posuny a hľadať ich spoločné prieniky. Prienikom roviny sa nájde spoločný bod dvoch obrázkov. Výsledkom týchto operácií je napríklad rektifikácia, ktorá spája dve snímky do jednej roviny.

4 Výsledky a záver

Vďaka algoritmom ako kalibrácia kamery a spracovanie riadkovej kamery bolo demonštrované základné spracúvanie obrazu pomocou knižnice OpenCV v jazyku Python vo vývojovom prostredí PyCharm. Bolo zistené, že knižnica OpenCV ponúka množstvo užitočných funkcií, ktoré priamo súvisia s teóriou bližšie uvedenou v Rešeršnej štúdii. Programovanie v jazyku Python je jednoduché a ľahko sa v ňom dá orientovať. Po úspešnej kalibrácii nasledovala simulácia riadkovej kamery, ktorá funguje na maticovom princípe a postupne spája jednotlivé riadky a snímky vstupného obrazu. Táto simulácia bola nahradená skutočnou riadkovou kamerou, z ktorej boli poskytnuté snímky v plnom rozlíšení. Týchto snímok však nebol dostatok a neboli v tej najlepšej kvalite, čo viedlo výpočtom, ktoré museli niektoré výchyľky odstrániť. Snímky boli iba dodané na spracovanie, nebola teda možnosť vyskúšať si manipulovanie s riadkovou kamerou a viacero parametrov sústavy bolo potrebné odhadnúť. Tieto snímky boli pomere strán, ktorý bolo potrebné napraviť. Na upravených snímkach boli testované algoritmy vykresľovania kontúr, korekcie objektu, rotácie snímky, rektifikácie, 3D spracovania modelu z dvoch snímok pomocou jazyka Python a vývojového prostredia Matlab, výpočtu matice homografie, porovnávacie algoritmy hľadajúce spoločné body... Meranie bolo časovo náročné a niektoré algoritmy nefungovali správne, čo vidno na priložených obrázkoch v priebehu práce. Viedlo to k vytvoreniu vlastného matematického modelu, ktorý vypočítal 3D model z jednej kamery. Následne bola rozobraná problematika snímania obrazu z dvoch kamier a vypočítané parametre potrebné k rekonštrukcie obrazu z dvoch kamier. Pri kvalitných fotografiách z rôznych uhlov pohľadu je možné dokázať, že výška je úmerná posunu odpovedajúcich si bodov. Plošné kamery ponúkajú jednoduchšie spracovanie obrazu, nakoľko je ľahšie ich kalibrovať. Pri kalibrácii riadkovej kamery je potrebné využívať viacero kalibračných objektov. Výpočet matice rotácie a vektoru translácie prebiehal manuálne, rovnako bola overená aj geometria, ktorá v meraní z dvoch snímok nastáva. Tento postup pomáha odstrániť výchyľky sústavy. Existuje však aj postup, ktorý by vedel odstrániť asymetriu výšok objektu pri väčších uhloch snímania, vyžaduje to však zložitejšie výpočty a je potrebné mať väčší počet snímok. V prílohách sa nachádzajú snímky zo simulácie riadkovej kamery, zmenšené snímky a výsledky 3D modelov.

5 Zoznam použitej literatúry

- [1] BLANÁRIK, I. *Epipolárna geometria* [online]. 2005 [cit. 2019-25-12]. Dostupné z: http://labss2.fiit.stuba.sk/TeamProject/2005/team05/doc/clanok_IB.pdf
- [2] *Camera Calibration*. docs.opencv.org. [online]. [cit. 2019-25-12]. Dostupné z: https://docs.opencv.org/master/dc/dbb/tutorial_py_calibration.html
- [3] *CCD vs CMOS*. www.teledynedalsa.com [online]. 2019 [cit. 2019-25-12]. Dostupné z: <http://www.teledynedalsa.com/en/learn/knowledge-center/ccd-vs-cmos/>
- [4] *Color Models*. www.archie-west.ac.uk [online]. 2012 [cit. 2019-25-12]. Dostupné z: http://www.archie-west.ac.uk/wp-content/docs/intel/ipp/ipp_manual/IPPI/ippi_ch6/ch6_color_models.htm
- [5] *Digitalizácia (obraz)*. sk.wikipedia.org [online]. 2013 [cit. 2019-25-12]. Dostupné z: [https://sk.wikipedia.org/wiki/Digitaliz%C3%A1cia_\(obraz\)](https://sk.wikipedia.org/wiki/Digitaliz%C3%A1cia_(obraz))
- [6] FOLTÍN, M. *Aplikácia pre podporu automatickej konverzie rastrových obrázkov*. Brno, 2017. Bakalárska práca. Fakulta informatiky, Masarykova Univerzita. Vedoucí práce RNDr. Jaromír Plhák, Ph.D.
- [7] GREŠ, T. *Řádkové kamery – typy a technologie* [online]. [cit. 2020-04-01] Dostupné z: <http://kamery.atesystem.cz/know-how/line-scan-velky-pruvodce-radkovymi-kamerami/dil-1-radkove-kamery-typy-a-technologie/?fbclid=IwAR0zIx-Pj83VPRzNgIKhCgvokjCcyP9KpBb7srBwKQY2D-Q4ArE-C-E220I>
- [8] HLAVÁČ, V. *Digitální obraz, základní pojmy* [online]. Praha: ČVUT v Praze [cit. 2019-25-12]. Dostupné z: <http://people.ciirc.cvut.cz/~hlavac/TeachPresCz/11DigZprObr/014DigitalImageCz.pdf>
- [9] KOVÁČ, O. *Teoretická analýza geometrickej kalibrácie kamery* [online]. 2019 [cit. 2019-25-12]. Dostupné z: <http://www.posterus.sk/?p=17190>
- [10] MAJUMDER, A. *Camera Calibration* [online]. [cit. 2019-25-12]. Dostupné z: <https://www.ics.uci.edu/~majumder/docs/cameracalib.pdf>
- [11] *Pinhole camera model*. en.wikipedia.org [online]. [cit. 2019-25-12]. Dostupné z: https://en.wikipedia.org/wiki/Pinhole_camera_model
- [12] *Princip kontrastu*. cs.wikipedia.org [online]. 2019 [cit. 2019-25-12]. Dostupné z: https://cs.wikipedia.org/wiki/Princip_kontrastu
- [13] VESELÝ, M. *Odstráňte súdkovité skreslenie* [online]. 2019 [cit. 2019-25-12]. Dostupné z: <https://digi-foto.sk/retus/casto-vyuzivane-techniky-pri-uprave-fotiek/odstranenie-sudkoviteho-skreslenia/#diskusia>
- [14] *Videokamera*. sk.wikipedia.org [online]. 2016 [cit. 2019-25-12]. Dostupné z: <https://sk.wikipedia.org/wiki/Videokamera>
- [15] VRBIČANOVÁ, A. *3D skener*. Brno, 2018. Bakalárska práca. Fakulta strojního inžinýrství, Vysoké učení technické v Brně. Vedoucí práce Ing. Tomáš Spáčil.
- [16] ŠOLONY, M. *Identifikácia pohybu v priestore*. Brno, 2007. Bakalárska práca. Fakulta informačních technologií, Vysoké učení technické v Brně, Ústav informačních systémů. Vedoucí práce Ing. Igor Potůček, Ph.D.

Prílohy



Príloha č. 1: *Vzor meraného objektu odfoteného plošnou kamerou – pohľad 1*



Príloha č. 2: *Vzor meraného objektu odfoteného riadkovou kamerou – pohľad 1*



Príloha č. 3: *Vzor meraného objektu odfoteného plošnou kamerou – pohľad 2*



Príloha č. 3: *Vzor meraného objektu odfoteného riadkovou kamerou – pohľad 2*



Príloha č. 4: *Meraný objekt odfotený horným riadkom kamery*



Príloha č. 5: *Meraný objekt odfotený stredným riadkom kamery*



Príloha č. 6: *Meraný objekt odfotený spodným riadkom kamery*